Computational Methods for High-Throughput Metabolomics

Ph. D. Thesis

submitted to the

Faculty of Technology,
Bielefeld University, Germany

for the degree of Dr. rer. nat.

by

Nils Hoffmann

May 26th, 2014

Referees: Prof. Dr. Karsten Niehaus
Prof. Dr. Jens Stoye

Printed on non-aging paper according to DIN-ISO 9706.
Gedruckt auf alterungsbeständigem holz- und säurefreiem Papier nach DIN-ISO 9706.

—To those who left too soon—

Zusammenfassung

Die immer häufiger werdende routinemäßige Anwendung analytischer Technologien in der Biologie und Biochemie zur quantitativen und qualitative Bestimmung kleiner Moleküle in biologischen Organismen hat in den letzten Jahren zu einem immer größer werdenden Bedarf an hochautomatisierten Verfahren zur Prozessierung und Analyse, sowie zum Vergleich großer Probenanzahlen geführt. Die bekanntesten Technologien in diesem Bereich sind die Chromatographie, die zur Trennung komplexer chemischer Gemische nach Molekülgröße oder -ladung, oder anderer Eigenschaften eingesetzt wird, sowie die häufig daran gekoppelte Massenspektrometrie, die das Masse-zu-Ladungsverhältnis von Ionen und Ionenfragmenten der zuvor chromatographisch getrennten Moleküle, sowie deren Intensität bestimmt.

Eine große Herausforderung bei diesen Hochdurchsatzmethoden ist die automatische Extraktion von charakteristischen Eigenschaften und die Quantifizierung der chemischen Verbindungen in den gemessenen Proben und deren zuverlässige Zuordnung zwischen mehreren Messungen für quantitative Vergleiche und statistische Analysen.

Das Hauptziel dieser Arbeit ist die Entwicklung und Evaluation von skalierbaren und robusten Methoden zur hochautomatisierten Prozessierung sehr vieler Messungen. Von besonderer Bedeutung ist hierbei der Vergleich verschiedener Messungen, um Gemeinsamkeiten und Unterschiede zwischen diesen im Kontext der Metabolomik zu finden; der Disziplin, die sich mit der Untersuchung und Charakterisierung kleiner Moleküle in biologischen Organismen beschäftigt.

In dieser Arbeit werden neue Algorithmen zum automatischen Abgleich von Peak- und Profilbasierten Daten aus ein- und zweidimensionalen Gaschromatographie-Massenspektrometrieexperimenten unter Zuhilfenahme der Retentionszeit beschrieben. Diese werden umfassend anhand öffentlich zugänglicher Datensätze von biologischer Relevanz gegen bereits etablierte Algorithmen verglichen.

Die zur Entwicklung der Algorithmen verwendete Programmbibliothek MALTCMS, sowie die grafische Benutzeroberfläche MAUI werden im weiteren Verlauf der Arbeit

vorgestellt. Die Anwendung beider wird mit Hilfe anschaulicher Beispiele exemplarisch dargestellt.

Abstract

The advent of analytical technologies being broadly and routinely applied in biology and biochemistry for the analysis and characterization of small molecules in biological organisms has brought with it the need to process, analyze, compare, and evaluate large amounts of experimental data in a highly automated fashion. The most prominent methods used in these fields are chromatographic methods capable of separating complex mixtures of chemical compounds by properties like size or charge, coupled to mass spectrometry detectors that measure the mass and intensity of a compound's ion or its fragments eluting from the chromatographic separation system.

One major problem in these high-throughput applications is the automatic extraction of features quantifying the compounds contained in the measured results and their reliable association among multiple measurements for quantification and statistical analysis.

The main goal of this thesis is the creation of scalable and robust methods for highly automated processing of large numbers of samples. Of special importance is the comparison of different samples in order to find similarities and differences in the context of metabolomics, the study of small chemical compounds in biological organisms.

We herein describe novel algorithms for retention time alignment of peak and chromatogram data from one- and two-dimensional gas chromatography-mass spectrometry experiments in the application area of metabolomics. We also perform a comprehensive evaluation of each method against other state-of-the-art methods on publicly available datasets with genuine biological backgrounds.

In addition to these methods, we also describe the underlying software framework MALTCMS and the accompanying graphical user interface MAUI, and demonstrate their use on instructive application examples.

Contents

List of Figures

List of Tables

Preface

Parts of this thesis have previously been published in a number of publications (Hoffmann et al. 2012; Hoffmann and Stoye 2012; Hoffmann et al. 2014). These parts were rearranged and substantially extended for the present work.

This includes the overview of existing frameworks for Metabolomics based on GC-MS and GC×GC-MS data covered in Hoffmann and Stoye (2012) in Chapters 1, 3, and 4. The derived application pipelines for MALTCMS are mentioned in Appendix A. The peak and raw data alignment algorithms BIPACE and CEMAPP-DTW were first described in Hoffmann et al. (2012) for GC-MS data. The algorithms are described in Chapter 3. The description of BIPACE has been substantially extended in this work. BIPACE 2D, for the alignment of peaks from GC×GC-MS data, based on BIPACE, was recently published (Hoffmann et al. 2014). It is covered in Chapter 4. The supplementary material of the BIPACE and BIPACE 2D publications is presented for completeness in Appendices B and C.

All links to internet resources such as websites or software downloads mentioned in this work have been checked and accessed between November 20th 2013 and January 12th 2014, unless mentioned otherwise.

1

Introduction

Metabolomics, the systematic study of the biochemistry of small molecules in biological organisms, has seen a rapid development of new technologies, methodologies, and data analysis procedures during the past decade. The development of fast gas- and liquid-chromatography devices coupled to sensitive mass-spectrometers, supplemented by the unprecedented precision of nuclear magnetic resonance for structure elucidation of small molecules, together with the public availability of database resources associated to metabolites and metabolic pathways, has enabled researchers to study the full collection of metabolites in different organisms, their *metabolome*, in a high-throughput fashion. Other *omics* technologies have a longer history in high-throughput applications, such as next generation sequencing for genomics, RNA microarrays for transcriptomics, and mass spectrometry methods for proteomics. All of these together give researchers a unique opportunity to study and combine multi-omics aspects, forming the discipline of *systems biology* in order to study organisms simultaneously at multiple scales and from different perspectives.

Like all other *omics* technologies, metabolomics data acquisition is becoming more reliable and less costly, while at the same time throughput is increased. Modern time-of-flight mass spectrometers are capable of acquiring full scan mass spectra at a rate of 500Hz from 50 to 750 m/z and with a mass accuracy <5 ppm with external calibration. At the opposite extreme of machinery, Fourier-transform ion-cyclotron-resonance (FTICR) mass spectrometers coupled to liquid chromatography for sample separation reach an unprecedented mass accuracy of <1 ppm m/z and very high mass resolution (Miura et al. 2010). These features are key requirements for successful and unique identification and characterization of unknown metabolites. Coupled to chromatographic separation devices, these machines create datasets ranging in size from a few hundred megabytes to several gigabytes per run. While this is not a severe limitation for small scale experiments, it may pose a significant burden on projects that aim at studying the metabolome or specific metabolites of many specimens and replicates, for example in medical research studies or in routine diagnostics

applications tailored to the metabolome of a specific species, such as the human (Wishart et al. 2009).

Thus, there is a need for sophisticated methods that can treat these datasets efficiently in terms of computational resources and which are able to extract, process, and compare the relevant information from these datasets and provide consistent and reliable results.

In this thesis, we describe such methods, addressing specifically the problems of peak and chromatogram alignment in one- and two-dimensional gas chromatography-mass spectrometry. The methods, among others for preprocessing, comparison, and annotation, are embedded into the software framework MALTCMS that we present in the later chapters of this work. MALTCMS is supplemented by the graphical user interface application MAUI for interactive exploration, processing and analysis of data from metabolomics experiments.

The remainder of this thesis is structured as follows: Chapter 2 introduces the reader to the discipline of metabolomics and gives an overview of the currently available and routinely applied analytical platforms. We further discuss the necessary and desirable features of a software framework for metabolomics data preprocessing based on GC-MS and comprehensive two-dimensional gas chromatography-mass spectrometry (GC×GC-MS) coupled to single-dimension detectors (flame/photo ionization, FID/PID) or multi-dimension detectors (mass spectrometry, MS). We therefore define a typical workflow for automatic data processing of metabolomics experiments and discuss available methods within each of the workflow's steps.

In Chapter 3, we compare the features of publicly available Open Source frameworks for GC-MS and present two methods for the peak and chromatogram alignment problems for GC-MS data, BIPACE and CEMAPP-DTW. The methods are evaluated against another state-of-the-art method on two representative datasets. Supplementary material for the evaluation is provided in Appendix B.

We then compare available Open Source frameworks for GC×GC-MS in chapter 4. We also describe a novel peak finding method based on the continuous wavelet transform. The problem of peak alignment in GC×GC-MS is addressed by our method BIPACE 2D, that is introduced and thoroughly evaluated against three other state-of-the-art methods and their variants on four different datasets. We provide additional supplementary material for the evaluation in Appendix C.

The methods are available in the Open Source software framework MALTCMS, that was developed during the author's work on this thesis. We describe MALTCMS in Chapter 5. It is tailored for use by domain experts and bioinformaticians who want to automate their metabolomics workflow with repeatable and auditable configurations. As a supplement to MALTCMS and for easier accessibility for novice, as well as expert users, we developed the modular graphical user interface application MAUI. The architecture and main functionality of MAUI is described in Chapter 6.

We summarize and discuss the results of this thesis in Chapter 7, before we finally give an outlook on the application and further development of MALTCMS and MAUI for high-throughput metabolomics.

In Appendix A, we additionally describe two pipelines for metabolomics analyses based on MaltcmS: ChromA, which is applicable to GC-MS, and ChromA4D, which is applicable to data from GC×GC-MS experiments. We show how to set up, configure and execute each pipeline using instructional datasets. These two workflows include the typical steps of raw-data preprocessing in metabolomics, including peak-finding and integration, peak-matching among multiple replicate groups and tentative identification using mass-spectral databases, as well as visualizations of raw and processed data. In the same appendix, we also give practical application examples of MaltcmS and Maui in the area of optimization of plant biomass production as a source of renewable energy and in the study of torpor, a state of metabolic suppression used for energy conservation in mice.

2

Background

We begin this chapter with a short introduction and review of *metabolomics* and its relation to the other major *omics* techniques: *genomics*, *transcriptomics*, and *proteomics*. We then describe the analytical methods used to study the metabolome in different organisms. The chromatographic methods used for the separation of complex mixtures of metabolites are introduced in Section 2.2, before we discuss mass spectrometry and briefly other detection methods that allow quantification of the metabolites separated by chromatography in Section 2.3.

Section 2.4 describes the different combinations of chromatography and mass spectrometry used in current analytical chemistry and metabolomics experiments. These hyphenated methods enable the separation and analysis of complex biological samples, a key requirement in metabolomics. In Section 2.5, we give a brief introduction into the terminology used in analytical chemistry and metabolomics with respect to these hyphenated methods.

We finally define a prototypical workflow for experimental metabolomics and explain the required steps in Section 2.6 and discuss available Open Source software implementations for the individual steps. The definition of this workflow will serve as a basis for the more specific workflows that are discussed in Chapters 3 and 4.

2.1. Metabolomics

The *metabolome* of a living organism comprises the entirety of molecules that act as substrates, intermediates, or products of biochemical reaction pathways (Nielsen and Jewett 2007). These molecules are called *metabolites*. *Metabolomics* as a term describing the associated scientific discipline involved with the study of metabolites was first coined by Oliver et al. (1998) in the context of functional genomics analysis of yeast. The scope of metabolomics is the elucidation of the functional phenotype of cells (Fiehn 2002) and the role that metabolites play in it. This observable phenotype is a result of the interplay of the *genome*, the *transcriptome*, the *proteome*, and, through

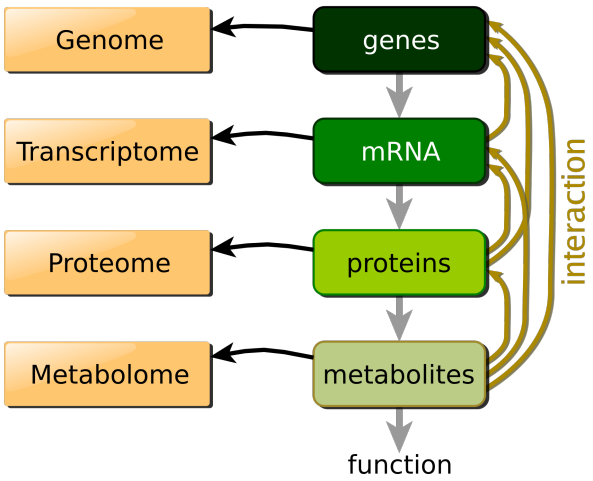


Figure 2.1.: A simplified model of the flow of information within the different *omics* levels. Adapted from Goodacre (2005).

various feedback interactions, the *metabolome* (see Figure 2.1). Many of the early advances in *metabolomics* originate in the field of biochemistry that studies the chemical reactions of metabolites which are mediated and catalyzed by enzymes (proteins), and that build parts of the complete metabolic network of an organism's cells.

The Calvin cycle in photosynthetic plants (Bassham, Benson, and Calvin 1950) and the Krebs cycle in aerobic organisms (Baldwin and Krebs 1981; Meléndez-Hevia, Waddell, and Cascante 1996) are prime examples of the early work required to elucidate metabolic reaction paths and of their important role in present-day *metabolomics*. However, back then the connection of these reactions to the genome was largely unknown. In order to reveal these connections, and their dynamic interaction, the data from different *omics* technologies need to be combined. First and foremost, genomic sequencing experiments (*genomics*) are employed to elucidate the genetic repertoire of an organism or cell (Fleischmann et al. 1995). This static knowledge is then supplemented by the dynamic information captured by gene expression experiments (*transcriptomics*) that help to determine the expression levels of genes that are influenced by external or internal perturbations, such as varying experimental conditions, at a given time (Lockhart et al. 1996). *Proteomics* adds the next layer of information (Shevchenko et al. 1996), identifying proteins as products of transcription and translation and their abundance and thus providing data on the cell machinery that is available for processing of substrate metabolites. Finally, *metabolomics* helps to determine the amounts of substrates, intermediates, and products in the cell under these conditions and is thus vital in assessing its dynamic activity. This integration of multiple *omics* techniques, with the aim to better understand the dynamic state of a cell, lead to the concept of *systems biology* (Mesarović 1968; Fiehn 2002; Sumner, Mendes, and Dixon 2003).

2.1.1. Challenges

Metabolites are very diverse in size and chemical functionality, ranging from amino acids, nucleotides, fatty acids, and ketones, to large polymer sugars and hormones. It is therefore impossible to analyze all metabolites present in a sample with a single analytical technology (Sumner, Mendes, and Dixon 2003). Therefore, different methods are applied for the separation of the metabolites contained within a sample, namely gas chromatography (GC) and liquid chromatography (LC) (see Section 2.2). These separation methods are often combined with different sensitive detectors, like mass spectrometers (see Section 2.3). This combination is termed *hyphenation* and the application of hyphenated methods is state-of-the-art in modern metabolomics (Dunn and Ellis 2005).

The concentrations of metabolites in biological samples can vary over up to nine orders of magnitude, and significant biological variation is also present between samples (Sumner, Mendes, and Dixon 2003). Thus, very sensitive detectors with a linear response over the range of possible concentrations (dynamic range) are required for quantitative applications, in addition to sophisticated statistical methods to handle the biological variation. Furthermore, specialized sample preparation protocols are often required to extract and reliably quantify metabolites that only occur in very small concentrations (Harrigan and Goodacre 2003, Chapter 1).

A further pressing issue in metabolomics is the identification of unknown metabolites, but the advent of mass spectrometers with very high mass resolution has opened new opportunities for computational methods that aid in the determination of metabolite sum formula and structure candidates (Neumann and Böcker 2010). In combination with nuclear magnetic resonance (NMR) (see Section 2.4) and other new spectroscopic technologies, these methods may pave the way for semi-automatic structure elucidation of unknown metabolites in the future.

2.1.2. Variants

Metabolomics as a field unifies different approaches to analyze and quantify metabolites in biological samples. The most complete approach is *comprehensive metabolomics* where as many metabolites as possible are identified and quantified with different analytical methods in order to gain a broad overview of the metabolism of the subject of study. However, this is also the most laborious and expensive variant employed in metabolomics. Thus, other variants focus on a more concise subset of metabolites and analytical methods.

In the context of biomarker detection, the term *metabolic fingerprinting* is often used to indicate that the presence or absence of a specific metabolite, or a small selection thereof, is used for disease indication and monitoring (Harrigan and Goodacre 2003, Chapter 1). If the fingerprinting is conducted using biofluids of human origin, it is often called *metabonomics*.

The last variant that is frequently applied is *metabolic profiling*. Here, a large selection of metabolites, usually those associated with particular biochemical pathways, are

qualitatively and quantitatively analyzed. Profiling is usually a *targeted* approach, where the metabolites under consideration are known beforehand.

All of these methods can in principle be performed without knowledge of the identities of the metabolites under study. Such *non-targeted* methods mainly use statistical methods to infer correlations of metabolite abundances across sample conditions (Aura et al. 2008; Koal and Deigner 2010), which may lead to the discovery of unknown metabolic intermediates or products.

Some examples where metabolomics methods are applied today are given in the following section.

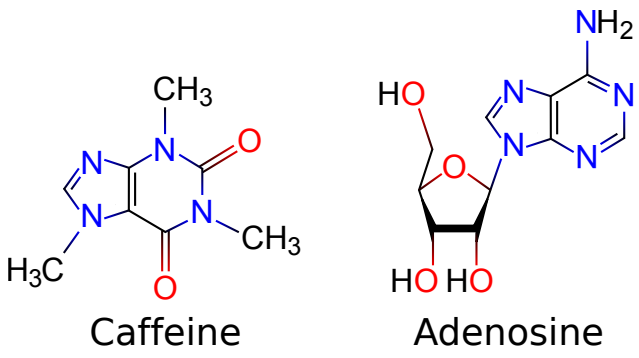


Figure 2.2.: Caffeine and Adenosine.[a]

a. Public Domain. Source: Wikimedia Commons, http://en.wikipedia.org/wiki/File:Caffeine_and_adenosine.svg

2.1.3. Applications

Xenobiotics, Toxicity, and Individualized Medicine Caffeine is a *secondary metabolite* of the cultivated plants *Coffea arabica* and *Coffea canephora*. Secondary metabolites are usually not essential to an organism's survival, but help it in many different ways, such as deterring herbivores and carnivores (e.g. alkaloids and terpenoids), or by confining the uncontrolled growth of bacteria (fungal antibiotics) in its environment. In contrast, *primary metabolites* are crucial for cell growth, reproduction, and development. Caffeine is a stimulating, psycho-active alkaloid drug that is frequently consumed by many humans. Since it is not synthesized by humans, it is termed a *xenobiotic* metabolite, when ingested. Caffeine is an antagonist of adenosine, blocking the adenosine receptors of nerve cells in the human brain due to its related structure (see Figure 2.2). Caffeine is generally attributed to increase alertness and attention, as well as to decrease fatigue. The xenobiotic metabolism in man, mediated by Cytochrome P450 1A2, rapidly demethylates caffeine (Arnaud et al. 1980) into four products (Tang-Liu, Williams, and Riegelman 1983) which are further metabolized and finally excreted in urine, so that toxic or even lethal doses can hardly build

up through the normal consumption of coffee. Other organisms, with a different xenobiotic metabolism may already be fatally intoxicated by small doses of caffeine.

The study of xenobiotics and their metabolized products is of vital interest for the assessment of toxicities of commercially produced chemicals and drugs in man and environment (Lahl and Hawxwell 2006). However, for many chemicals it is not known where they are metabolized and what their intermediate products are. Weckwerth (2011) shows the importance of interlinking genomics (high-throughput sequencing data), proteomics, and transcriptomics data with metabolomics data in order to locate and close gaps in biochemical pathways. This is a requirement for the prediction of the toxic potential of chemicals in man and other organisms. Furthermore, deeper knowledge in this area also allows to assess the suitability of native metabolites as disease biomarkers and of novel chemicals as potential drugs for specific and individual disease treatment (Weston and Hood 2004; Greef, Hankemeier, and McBurney 2006; Baraldi et al. 2009). Potential targets for the discovery of novel drugs are plants, with an estimated number of 200.000 metabolites (Fiehn 2002), most of which have yet to be identified.

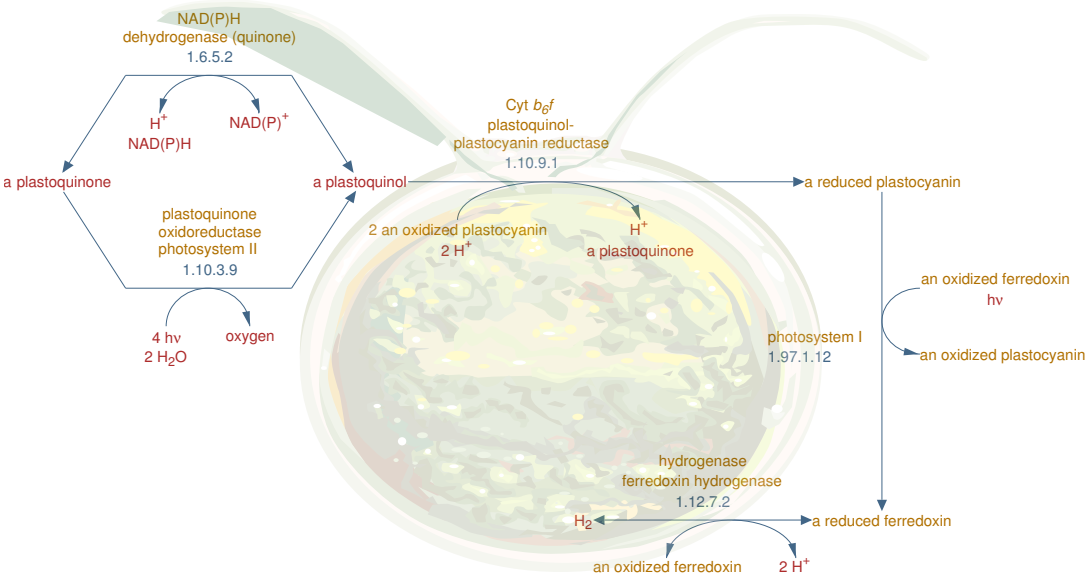


Figure 2.3.: H_2 Production Pathway of the fresh water algae *Chlamydomonas reinhardtii* from the BioCyc Database (Caspi et al. 2012). Under anaerobic conditions, induced by sulfur depletion, the green algae *C. reinhardtii* produces H_2 in the chloroplast during photosynthetic activity. Enzymes are represented by their Enzyme Commission (EC) number and common name. $\hbar\nu$ indicates the exposition to photons from sunlight or artificial illumination. Background: Single *C. reinhardtii* cell[a].

a. with permission for non-commercial use from http://www.pflanzenforschung.de

Xanthan Production The γ-proteobacterium *Xanthomonas campestris pv. campestris* (XCC) B100 produces the polysaccharide xanthan that is industrially used in food and cosmetics products as a viscosifier (Schatschneider et al. 2013). Therefore, an optimized production yield of xanthan with the same environmental parameters would lead to a more cost-effective product. In order to optimize the cultivation environment or the organism, or both, a sound knowledge of biochemical pathways of XCC is required. This includes genes, transcripts, proteins and metabolites, as well as their interactions, in order to find targets for yield optimization. Genetic variants of XCC and different environmental conditions can then be tested in their metabolic response and production of xanthan against the wildtype with metabolomics techniques.

Hydrogen Biofuel Production The fresh water algae *Chlamydomonas reinhardtii* (*C. reinhardtii*) produces hydrogen (H_2) under anaerobic conditions induced by sulfur depletion (Melis et al. 2000; Hemschemeier et al. 2008; Matthew et al. 2009; Doebbe et al. 2010). H_2 is an important starting point for biofuel production from renewable sources. One of the advantages of *C. reinhardtii* over crop plants for biofuel production are the smaller amount of space needed to grow them on with a comparable energy balance. Their cultivation tanks have no requirement for arable farmland that would otherwise be used for food production. Additional advantages of the algae are the feasibility of its cultivation in sea and waste water, and the high, year-round harvesting frequency (Schenk et al. 2008). The optimization of H_2 production in *C. reinhardtii* can again be assessed using metabolomics techniques, by monitoring and comparing the amounts of metabolites that are directly or indirectly involved in the H_2 production pathway (see Figure 2.3) between different genetic variants and environmental conditions (Doebbe et al. 2010). A dataset from such an experiment was used for the evaluation of the algorithm described in Section 4.3.

2.2. Chromatography

Chromatography (from the Greek words for *color* and *to write*) is generally defined as the separation of complex mixtures of analytes, e.g. metabolites, into their components. A chromatographic separation requires a *mobile phase* (gas or liquid), termed the *eluent* or solvent, and a *stationary phase*. The analytes suspended within the eluent exhibit adhesive interactions (*adsorption*) with the stationary phase, mediated by the solvent, while being moved along the stationary phase by a directed gradient.

In column chromatography, the stationary phase is usually located inside a column, either as a thin coating on the column wall, or as larger particles that are packed inside the column. In paper chromatography, the stationary phase is usually a porous filtration paper.

Adsorption chromatography builds the foundation of modern gas and liquid column chromatography. Its invention and description is attributed to Michail Zwet (also known as *Tswett*) and was used by him for the separation and characterization of plant pigments, like chlorophyll and carotenoids, in the early 1900s (Zwet 1906). Zwet used

manually packed columns with calcium carbonate as adsorbent material, flowing the plant pigments in liquid solution through the column to separate them. He also introduced the terms *chromatogram* and *chromatographic method* for the detected result of the separation and the process as a whole.

Another important foundation for modern column chromatography with liquid mobile phases was the invention of *partition chromatography* by Archer Martin and Richard Synge, who were awarded the Nobel prize in chemistry in 1952 for their contribution to the field. They added a liquid phase to the adsorbing material coating their columns to improve column selectivity and peak resolution originally for gas chromatography, but ultimately providing the foundation for modern high-performance liquid chromatography (HPLC) (Lovelock 2004). Additionally, they laid the foundation for models of peak capacity and separation performance for column chromatographic systems by introducing the theoretical plate model.

In the following sections, we will give an overview of the chromatographic methods that are routinely used in metabolomics. A more comprehensive overview of different methods for metabolite extraction, separation, especially of polar analytes, quantification and identification in metabolomics can be found in the books of Weckwerth (2007) and Harrigan and Goodacre (2003).

2.2.1. Gas Chromatography

GC is a variant of column chromatography, with an inert gas (e.g. Helium or Nitrogen) as the mobile phase. The columns are typically either capillary columns with a coating of polysiloxanes or packed columns with a solid or liquid stationary phase, allowing for a large range of selectivity for the separation of analytes with polar, hydrophilic, or other physicochemical properties. For complex mixtures of analytes, a common use-case in metabolomics, capillary columns offer better peak capacity and therefore better resolution of peaks over packed columns. Figure 2.4 shows a schematic of a gas chromatograph coupled to a mass spectrometer as the detector. The column is placed inside a temperature programmed oven. During a chromatographic separation, the oven's temperature profile can be changed to reduce the adhesion of analytes to the stationary phase. When the sample is injected, it is moved through the column by the gas flow, where the analytes interact with the stationary column. If an analyte interacts scarcely with the stationary phase, it will elute from the column before analytes that exhibit a higher interaction due to adsorption. Analytes exiting the chromatograph are transferred to a detector. In metabolomics, common detectors are flame ionization detectors (FIDs) and mass spectrometers (Tian et al. 2008; Koek et al. 2006; Dettmer, Aronov, and Hammock 2007).

FIDs are used for the detection and quantification of organic analytes (McWilliam and Dewar 1958). The analytes are combusted using hydrogen gas and an oxidant (e.g. oxygen). The difference in electric current between the positively charged outlet electrode and the negatively charged collector electrode attracts reduced ions exiting the flame. The ion signals are amplified and integrated to produce a time resolved

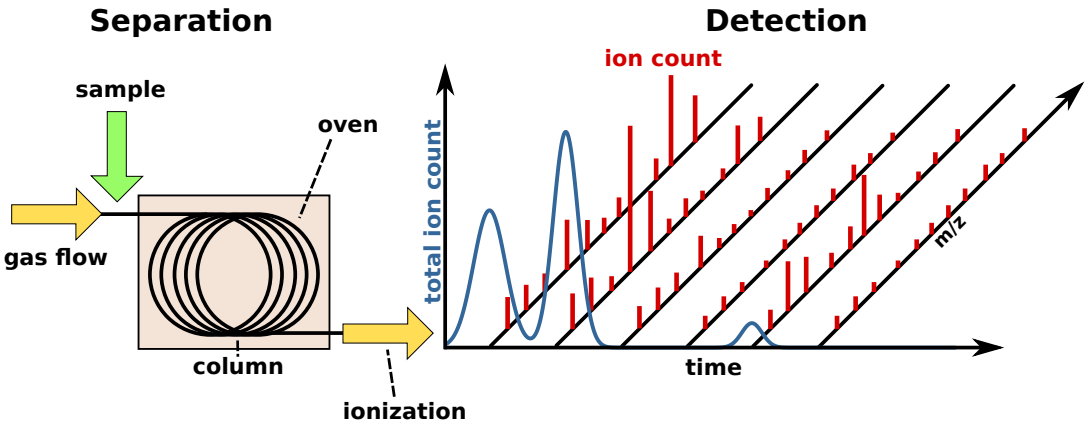


Figure 2.4.: Schematic of a GC-MS device. The sample is injected into the system and moved over the column by the inert carrier gas flow through a heated oven. After exiting the column, the eluting compounds are ionized and transferred to the detector. The response of the detector is recorded over time, here as a sequence of mass spectra.

response value that is related to the concentration of reduced carbon atoms pyrolized at the time of measurement.

We will discuss the different mass spectrometry methods available for GC in Section 2.3.

2.2.2. Liquid Chromatography

Liquid chromatography (LC) is also a variant of column chromatography, but with a liquid mobile phase. Today, most LCs are operated at very high pressures (HPLC), in order to achieve a better separation of the analytes within 30 to 60 minutes of an experiment run. The columns used in LC are generally much shorter and more compact than those used in GC. They are filled with porous materials that are coated with solid or liquid material, exhibiting different adsorption characteristics. LC columns are generally operated at lower temperatures than those used in gas-chromatography. Here, the adsorption is often regulated via a varying solvent gradient that allows to vary the selectivity from polar to non-polar analytes, in order to achieve a better separation. One challenge in LC is the transfer of eluting analytes to the detector. Usually, the solvent material has to be removed before or during ionization of the analytes. Ionization methods like electrospray ionization (ESI) and atmospheric pressure chemical ionization (APCI) provide a convenient coupling of LC and mass spectrometry (MS) (see Section 2.3.1 for more details). LC also covers a much higher range of analyte masses, enabling the separation and analysis of small metabolites, as well as larger peptides and even proteins.

2.2.3. Two-dimensional Chromatography

Chromatography with one separation column often encounters problems for the complex samples measured in metabolomics experiments. Here, the peak separation is often not optimal, especially for chemically closely related analytes. These co-eluting analytes can often be separated by introducing a second column with different characteristics, such as polarity. In practice, the two columns are coupled by a modulator or switching pump with a defined volume that is filled with eluate from the first column and released onto the second column within a fixed time interval (Mondello et al. 2008).

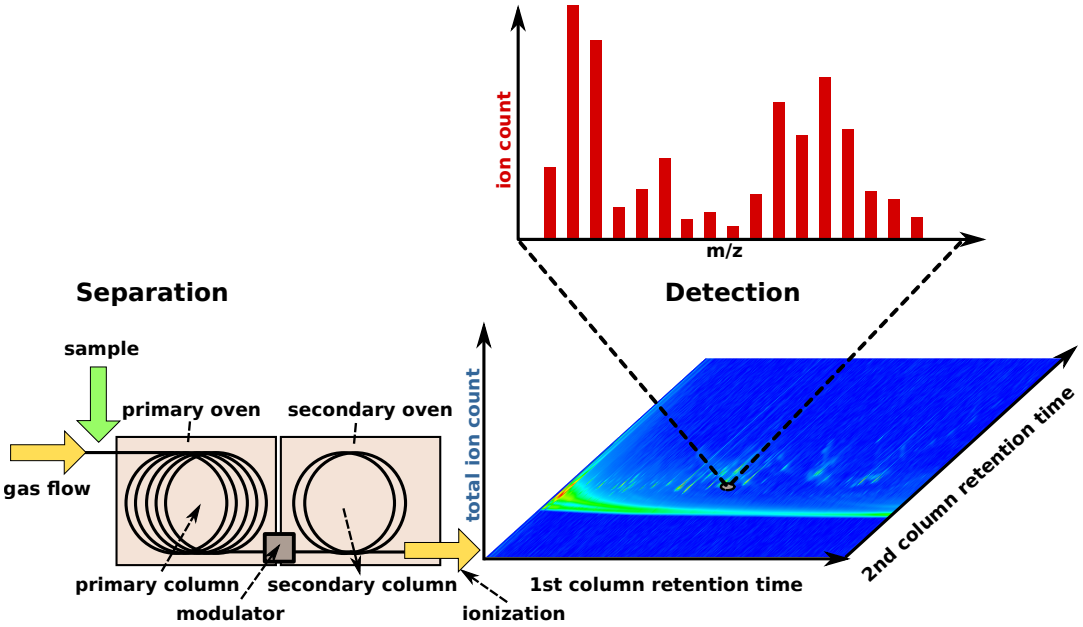


Figure 2.5.: Schematic of a GC×GC-MS device. The sample is injected into the system and moved over the first column by the inert carrier gas flow through the primary oven. The eluting compounds are captured in a modulator for some time before being released onto the second separation column (modulation period) within the secondary oven (often contained within the first one). When the separated compounds exit the second column, they are ionized and transferred to the detector. The detector's response is recorded over time, in this case as mass spectra with two retention times. The first column retention time axis represents the time at the start of a modulation period on the first column, while the second column retention time is calculated from the local scan acquisition time since the start of a modulation.

A schematic of a comprehensive two-dimensional gas chromatography (GC×GC) device with a mass spectrometer as its detector is shown in Figure 2.5. The coupled column setup imposes a limit on the possible length and flow rate of the second column, since the mobile phase volume captured in the modulator has to traverse the second column within the fixed time interval. After exiting the second column, the analytes are transferred to a mass spectrometer for detection. Koek et al. (2011) show

that the improved peak capacity and lower detection limit in GC×GC-MS increase the number of biomarkers found when compared to GC-MS, which is especially helpful in the context of metabolomics.

2.3. Mass Spectrometry

The objective of mass spectrometry is to measure the mass and charge of ions as accurately and fast as possible. J.J. Thomson is generally attributed as the inventor of mass spectrometry, even though his initial work was focused on determining the nature of positively charged cathode rays. These rays were only later understood to be ions, and Thomson's work incidentally lead to the construction of the first mass spectrometer to study their nature in 1897 (Griffiths 2008). He was also the first to indirectly measure the mass of the electron via the charge-to-mass ratio and the charge that his refined apparatus could detect at the same time, earning him a Nobel Prize in Physics in 1906. At that time, the ions were being detected on a photographic plate, while today, detectors amplify and record the change in electric charge induced by the ion colliding with the detector surface. But it took another 80 years until ionization was sophisticated enough to also measure larger biomolecules, like complex sugars and proteins. The introduction of ESI by John Fenn, and of the principles of soft laser desorption ionization by Koichi Tanaka in the late 1980s opened the door for the routine application of MS in biology and biochemistry (Griffiths 2008).

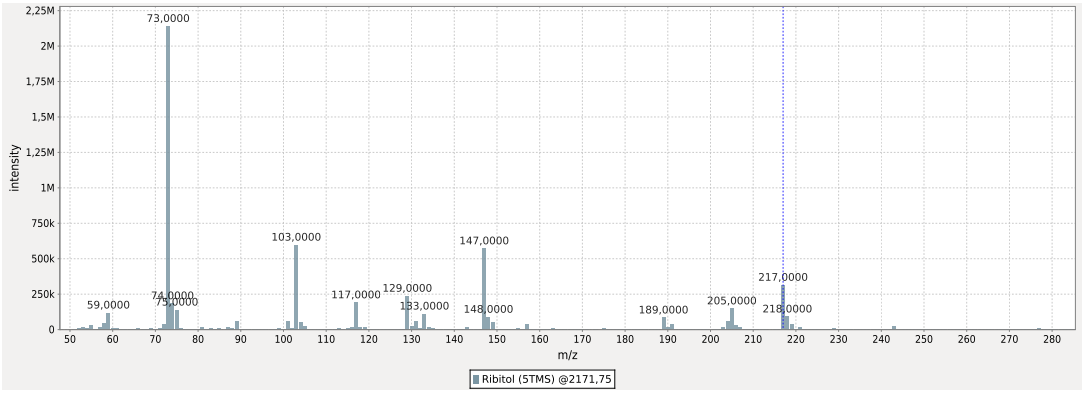


Figure 2.6.: Nominal mass electron ionization mass spectrum of Ribitol (5TMS). Annotation was performed against the Golm Metabolome Database (Hummel et al. 2007).

Mass Spectrum A *mass spectrum* consists of pairs of ion mass-to-charge ratio and non-negative intensity (sometimes called *count*) for the ions detected by the mass spectrometer. The *mass-to-charge* ratio m/z is a dimensionless fraction of multiples of the unified atomic mass (m_u, with unit Da), which is defined as $\frac{1}{12}$'th of the mass of the ^{12}C isotope of carbon, and of the charge number z, which is the number of

positive or negative charges of an ion. Thus, an M^+ cation (M for Molecule) has a charge number of $z = +1$ and an elementary charge of 1 e, where e is the absolute electric charge carried by a proton or electron. The masses that are measured by the mass spectrometer as m are those of the analyte's radical cation $M^{+\bullet}$ (the parent ion), that is generated within the ion source of the mass spectrometer from the analyte molecule in positive ionization mode by removing one electron, when electron ionization (EI) is used (see Section 2.3.1 for details). It is not unusual, especially if high-energy ionizations are used, that the parent ion is hardly detectable or not present at all. This happens if the ionization energy transferred to an analyte is large enough to break covalent bonds within it, leading to fragmentation and often also to rearrangement reactions (McLafferty 1959). However, these fragments carry a lot of information about the original analyte in them, and if the masses are measured with sufficient resolution and accuracy (see Section 2.3), they can be used to infer the original elemental composition of the parent ion as a sum formula, or even to predict multiple candidate structures for the (predicted) parent ion (Hufsky et al. 2012).

Figure 2.6 is an example of a nominal mass spectrum acquired using EI in positive mode of the internal standard Ribitol that is commonly used in metabolomics for peak area normalization (Barsch, Patschkowski, and Niehaus 2004). Ribitol, as all polar organic analytes, requires a prior derivatization with trimethylsilyl (TMS) reagents to make it volatile. TMS is used primarily on analytes containing hydroxy- or carboxy-groups, such as alcohols and carboxylic acids, substituting a hydrogen and binding with the oxygen. It is often complemented by the addition of methoxylamine hydrochloride in pyridine solution to open the cyclic isomers of sugars. TMS can then substitute the hydroxy groups of the sugar. The mass of derivatized Ribitol, in this case with five TMS groups, is expected at m/z 513. But due to the use of EI, the parent ion is not measurable (and consequently not shown in Figure 2.6). Thus, the mass spectrum only includes smaller fragments of Ribitol, including derivatization artifacts and column bleed contaminations, such as polysiloxanes at m/z 73 and 147.

Components A mass spectrometer consists of three basic parts (see Gross 2011, Chapter 2): the *ion source*, the *mass analyzer*, and the *detector*. One of the challenges in the coupling of chromatography and mass spectrometry is the transfer from the pressurized chromatography column to the high vacuum conditions that are prevalent in the mass spectrometer. Therefore, ion sources are usually preceded by an interface that mediates the transition from one system into the other and that transfers the analytes to the gas phase. The mass spectrometer is also connected to a digital computer to record the output of the detector for later processing and analysis, often involving the possibility to control the mass spectrometer's selectivity for certain ions and to repeatedly fragment them to obtain more structural information about the parent ions for MS^2 and MS^N applications (see Gross 2011, Chapter 9).

Resolution and Accuracy The performance of a mass spectrometer can be characterized by different numbers. First and foremost, the *mass resolution* of a mass

spectrometer is the smallest difference $\frac{m_2-m_1}{m_2}$ in m/z between two ions with masses m_1 and m_2, with $m_1 < m_2$, that still can be distinguished as two unique signals. Its inverse, the *resolving power*, is also often used to characterize the performance of a mass spectrometer. Some mass spectrometers have unit mass resolution, meaning that they can only distinguish equally charged ion signals that are at least one Da apart. The second performance measure is *mass accuracy*, the expected variation of a repeatedly measured m/z of an ion against its *true* m/z, measured in parts-per-million (ppm). For some mass analyzers, accuracy can decrease with increasing m/z. Optimally, a high *mass resolution* should always be complemented with a high *mass accuracy*, where the expected accuracy should be smaller than the smallest detectable m/z difference. Higher resolution mass spectrometers often acquire the mass spectra in *continuous* mode, which is later converted internally to *centroided*, corrected data.

Finally, the *scan rate* determines the maximum number of full scan mass spectra that a mass spectrometer can acquire within a second of operation. Modern instruments achieve a scan rate of more than 500 Hz, e.g. the LECO Pegasus 4D GC×GC-TOF-MS instrument (LECO Corp, St. Joseph, MI, USA) at unit mass accuracy, or the LECO Pegasus GC-HRT, which has a scan rate of 200 Hz at less than 1 ppm mass accuracy.

2.3.1. Ion Sources

The most commonly used ionization method used with GC instruments is electron ionization (EI). Analytes passing the EI source are ionized by an electron beam that is usually set to an energy of 70 eV. EI is a *hard* ionization method, as it leads to a fragmentation of the parent ion immediately after ionization. EI can also be used in combination with LC, however, this requires an intermediate step to remove the solvent material (Gross 2011, Chapter 5).

Chemical ionization (CI) is softer than EI in the sense that the ionization is not directly performed by an electron beam. Instead, analytes are ionized when they collide with molecules of a reagent gas (methane, isobutane, ammonia). In contrast to EI, the resulting protonated parent ion (usually $[M+H]^+$) is mostly stable. However, the analyte ion may also form adducts with the reagent gas used for ionization, which requires further care when interpreting mass spectra obtained after CI. CI can also be used for negative ionization (Gross 2011, Chapter 7).

The most commonly used ionization method for LC instruments is ESI. It operates at atmospheric pressure, and enables the transfer of small analyte molecules, as well as large molecules like proteins, from the liquid mobile phase to the gas phase. In ESI, the analytes and solvent are transferred through a charged capillary nozzle to form a spray, transferring the analytes in solution to the gas phase. After exiting the nozzle, the solvent is removed from the spray, before the analyte ions are then transferred to the mass analyzer (Gross 2011, Chapter 12).

2.3.2. Mass Analyzers

The mass analyzer separates ions based on their m/z ratio to allow specific detection of the individual ions and their abundance by the detector.

Quadrupole A quadrupole mass analyzer consists of four parallel metal rods with cylindrical or hyperbolic shape. The rods are pairwise oppositely charged with a mixture of alternating current (AC) and direct current (DC). By varying the AC frequency and voltages of both currents, it is possible to select ions with a defined mass and to move them along the elongation of the rods towards the detector. Ions that have a higher mass collide with the rods, while ions with lower mass are accelerated and ejected at the side of the rods. Quadrupole detectors can be used to measure ions up to 2000m/z, but only with 0.1 Da to 1 Da resolution.

Quadrupoles can also be modified to operate as ion traps, capturing ions of a defined mass within the rods, confined by electrical potentials at the entry and exit ends of the rods. These are often combined as triple quadrupoles, that allow to select ions in the first quadrupole stage, collide and fragment the ions with CI in the second quadrupole, and select fragment ions in the third stage, before transferring them to the detector. Thus, a triple quadrupole can be used for tandem mass spectrometry (MS/MS) applications (Gross 2011, Chapter 4) such as multiple reaction monitoring (MRM) (Kondrat, McClusky, and Cooks 1978) for the precise quantification of selected peptides and metabolites (Kitteringham et al. 2009).

Fourier Transform Ion Cyclotron Resonance Fourier transform ion cyclotron resonance (FT-ICR) combines very high mass resolution (1.0×10^{-5} Da to 1.0×10^{-6} Da) and accuracy of < 1 ppm. Since its invention in 1974 (Comisarow and Marshall 1974), it has been continuously refined and improved, by using stronger and larger superconducting magnets, as well as improved electric field generation. FT-ICR requires strong magnets to create a static magnetic field that is used to hold ions on a circular path within a miniature particle accelerator. The ions are accelerated by an oscillating electric field, with an orientation perpendicular to the magnetic field, until they reach their *cyclotron frequency*. All ions of the same m/z then move in phase and pass the electrodes used for detection with their *cyclotron* frequency as a *swarm*. Thus, for multiple ions, the detector readout is a linear combination of sine functions with different phase, frequency, and power. The individual ion masses and abundances can then be reconstructed from this *interferogram* by applying the Fourier transform (see Section 4.2.2 for a short overview and references) to them. Due to its dependence on strong, superconducting magnets, FT-ICR is rather expensive and requires dedicated laboratory rooms for secure operation.

Orbitrap The Orbitrap (Hu et al. 2005) shares the concept of moving ions on a circular path, based on their m/z and requires the Fourier transform to deconvolve the signals of different ions. However, it does not use a superconducting magnet and

is thus much cheaper to construct and maintain. It has mass resolution comparable to the FT-ICR, but with lower accuracy of 2 ppm to 5 ppm, and a mass range of up to 6000 Da. It operates by forcing ions into spiraling orbits around a central, spindle-shaped electron that is encased by an outer electrode that consists of two electrically insulated parts. The frequency of an ion orbiting the central electrode is a direct function of its mass and charge, thus, the m/z is reconstructed from the differentially measured current between the two parts of the outer electrode, when ion swarms move backwards and forwards along the central electrode with their characteristic frequencies. The resulting *interferogram* is again deconvolved similarly to FT-ICR to determine the ions' masses and abundances.

Time-Of-Flight Instruments

In time-of-flight (TOF) instruments, the ions are exposed to an electromagnetic field with fixed energy, accelerating them on their way to the field free flight tube. The field transfers the same amount of kinetic energy to every ion at the same charge, thus lighter ions with the same charge move at higher velocities, while heavier ions with the same charge travel at lower velocities, before they arrive at the electron detector. Modern TOFs are often equipped with a *reflectron* (reTOF) that acts as a focusing ion mirror in order to increase the flight tube length and to reduce the effect of flight time dispersion for ions with similar mass, with the result of an increased mass resolution. TOF detectors are relatively cheap to build, while they can cover an (almost) unlimited range of m/z values and can be tuned for accurate mass measurements (Vestal 2009; Gross 2011) (well below 10 ppm), and high spectra acquisition rate. They are also used for MS/MS applications (TOF/TOF).

2.3.3. Detectors

The ion analyzers transform physical properties of an ion (typically charge) into an electric signal. The strength of the signal correlates with the measured amount of ions detected in a short time span, but it is usually too weak to be processed directly. Therefore, detectors like the *Faraday cup*, *discrete dynode electron multipliers*, *channel electron multipliers*, *microchannel plates* and other methods have been developed to amplify the ion signals to currents, that are reliably measurable and convertable to ion intensities by an *analog-to-digital* converter (Gross 2011, Chapter 4). The different methods serve different purposes, such as to allow for linear signal response in a wide mass range, or to restrict the detector to a small size for better portability of the whole mass spectrometer.

2.4. Hyphenated Methods

The coupling of a chromatograph to a detector is termed *hyphenation*. Methods such as gas chromatography-mass spectrometry (GC-MS), liquid chromatography-mass

spectrometry (LC-MS), or comprehensive two-dimensional gas chromatography-mass spectrometry (GC×GC-MS) combine the separational capabilities of their chromatographic system with a mass spectrometer as a sensitive detector. The coupling is however non-trivial, since mass spectrometers operate at high vacuum conditions. In GC-MS, the coupling is comparatively easy, since only the gaseous mobile phase has to be removed before the volatile analytes are ionized and transferred to the mass analyzer. In LC-MS, the analytes are solvatized in the liquid mobile phase, which has to be carefully removed before or during ionization (see Section 2.3.1).

The first GC-MS devices to operate on another planet were onboard NASA's Viking I and II landers, that touched down on Mars in 1976[1]. Before then, laboratory GC-MS devices occupied the space of a room, while the Viking devices were confined to the size of a hat box with a maximum weight of 15 kg. Today, affordable and powerful benchtop devices are commonplace in laboratories around the world for routine analysis in diverse areas, including, but not limited to banned substance control (Moeller, Fey, and Wennig 1993), chemical warfare agents (Black et al. 1994), pesticide screening in environmental control (Benfenati et al. 1990), as well as metabolomics (Weckwerth 2011).

GC and GC×GC coupled to FID and MS detectors were used to determine the amount and composition of crude oil in water samples taken at different sites and in alternating depths from the gulf of Mexico after the Deepwater Horizon oil spill in 2010 (Reddy et al. 2011). Some of the more exotic usages of GC×GC-MS include the profiling of volatile organic compounds from decaying pig carcasses for forensic studies (Brasseur et al. 2012), while it is also applied for drug analysis and doping control (Kueh et al. 2003), as well as for metabolomics (Koek et al. 2011; Shellie et al. 2005; Pierce et al. 2006).

Other separation techniques like capillary electrophoresis (CE) for the separation of very polar analytes, and ion mobility spectrometry (IMS) have not been covered in this overview, although they are also used to cover parts of the metabolome that are otherwise inaccessible by means of other separation methods. Additionally, there exists a vast diversity of detectors available for coupling to a chromatographic system that were not covered here. Most prominently, NMR is a valuable tool for structure elucidation of unknown metabolites. However, it requires large amounts of analyte to operate, which can be problematic for substances that can not be easily isolated and purified in the necessary amounts from their biological source. A comparison of the most important hyphenated techniques and their application in metabolomics are described by Weckwerth (2011).

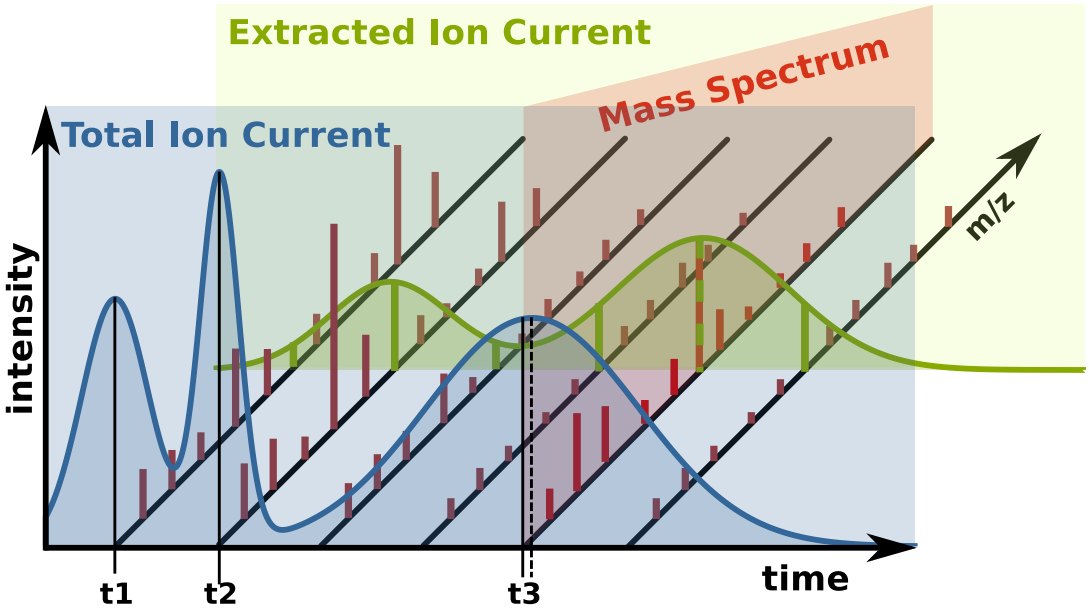


Figure 2.7.: Schematic structure of data in GC-MS and liquid chromatography-mass spectrometry (LC-MS). Peak heights are exaggerated for visualization purposes. Peaks at retention times t1 and t2 show overlapping behavior in the total ion current (TIC) (blue panel), but can be well separated as extracted ion currents (EICs) (green panel). The peak at retention time t3 illustrates that the TIC apex of a peak (dashed line) is not always sampled exactly by one mass spectrum (red panel).

2.5. Terminology for Data acquired with Hyphenated Methods

When a mass spectrometer is coupled to a chromatographic system, mass spectra are usually acquired at a fixed scan rate[2]. Thus, the data obtained from GC-MS or LC-MS experiments are sequences of mass spectra (see red panel in Figure 2.7), each with an associated time stamp, the *scan acquisition time*. Analytes exiting the chromatographic system show a time dependent abundance profile, starting with a low abundance, apexing at a maximum abundance, and ending with a low abundance. Such a profile is termed a *peak*. The scan acquisition time at the peak's apex is termed the *retention time* of the corresponding analyte. The full profile of an ion *count* or *current* at a specific m/z from the beginning of the MS acquisition until its end is termed an extracted ion current (EIC) (see green panel in Figure 2.7). If we sum, for each scan acquisition time along the m/z axis, all EICs at that specific time, we obtain the total ion current (TIC) (blue panel in Figure 2.7). Thus, the TIC often contains the sum of many weak ion currents, leading to a higher noise level than the individual EICs.

The bell shaped profile of a TIC or EIC peak is a result of the interplay of adsorption and resorption effects between the analyte, the stationary, and the mobile phase

1. http://appel.nasa.gov/2010/09/20/aa_3-9_f_history-html
2. Exceptions are data-dependent MS/MS fragmentations that may require more time than the inter-scan time between two consecutive regular mass spectral scans.

within the chromatographic system. In general, narrow peaks with a symmetric shape would be optimal, however in practice, the peak shape can vary for many reasons like non-optimal analyte concentration with respect to the column, or the temperature of the column. Peaks often exhibit a tailing behavior, meaning that the front has a steeper ascent than the descent of the profile following the apex. Thus, there is generally no simple analytical function that could model a typical peak shape.

Figure 2.7 shows that the peaks with retention times t1 and t2 overlap in the TIC, while they would be distinguishable in individual EICs. The peak with retention time t3 shows another aspect of the acquisition of mass spectra with a fixed scan rate. There is no guarantee that the true apex of a peak is sampled by a mass spectrum. Thus, it is necessary to have a high scan acquisition rate for quantification purposes, minimizing the risk of sampling a peak at only a few positions. Following the Nyquist-Shannon sampling theorem (Shannon 1949), the sampling frequency, here the scan acquisition rate, must be smaller or equal to one half of the frequency of the narrowest peak (with the highest frequency) in the chromatogram to avoid sampling aliasing artifacts. These artifacts would appear as artificial peaks in the sampled chromatogram (TIC or EIC) but would not be distinguishable from real peaks. Thus, for a scan acquisition rate of 100 Hz, one can sample peaks with a maximum width of 0.02 s in order to avoid sampling artifacts. These high scan acquisition rates are necessary in GC×GC-MS to ensure good resolution of chromatographic peaks due to the short second separation column (see Chapter 4 for more details).

For reasons of simplicity, the peak shape is often idealized as a Gaussian probability density function. A full chromatographic profile is thus the superposition of multiple Gaussians with different parametrizations (scale/standard deviation and mean). Alternative parametric peak models like the inverse Gaussian are used for improved modeling of tailing peak shapes (Hauschild et al. 2013). For quantification purposes, a peak's area is the *area-under-curve* that is obtained by integrating the peak from its beginning to its end. The area is usually corrected by subtracting the area of the estimated baseline function that models chemical and detector noise within the peak bounds. Analytically, the bounds of an ideal peak can be determined from its profile by finding local minima closest to the left and right of the peak's apex by inspecting the first and second order derivatives. In practice the peak profile often needs to be preprocessed to be smooth enough so that local noise in the profile does not influence the finding of peak apices and their start and end. Peak detection and integration are still active areas of research due to the complexity and differences involved with the various chromatographic and mass spectrometric technologies applied in metabolomics and proteomics research (Windig, Phalp, and Payne 1996; X. Zhang et al. 2005; Jonsson et al. 2005; Smith et al. 2006; Tautenhahn, Böttcher, and Neumann 2008; Fredriksson et al. 2009; Vivó-Truyols 2012).

In the remainder of this thesis, we will use *peak* as a synonym for a mass spectrum with additional one- or two-dimensional retention time information. Such a mass spectrum can either be the result of simply selecting the mass spectrum acquired closest to the actual peak's apex, or it can be the result of a deconvolution step that

separates overlapping ion signals from other peaks to receive a clean mass spectrum for the respective peak (Biller and Biemann 1974; Colby 1992; Stein 1999; Likić 2009).

2.6. A Typical Workflow for a Metabolomics Experiment

We have introduced hyphenated methods for the separation and detection of analytes in the previous sections for the analysis of samples from metabolomics experiments. There are however some challenges associated with these analytical methods that we will further elaborate within the steps of a typical workflow for a metabolomics experiment, based on one- and two-dimensional chromatography-mass spectrometry.

We will give a short overview on published methods for each step of the workflow that are available under an Open Source license, thus allowing researchers to examine their actual implementation details. This distinguishes these methods from applications that are only provided on explicit request, under limited terms of use, or that are not published together with their source code (Lommen 2009; Stein 1999), which is still often the case in metabolomics and may hamper comparability and reuse of existing solutions. Additionally, all software frameworks introduced in Chapters 3 and 4, that may model parts or the whole of such a processing pipeline, are available for all major operating systems such as Microsoft Windows, Linux, and Apple Mac OSx as standalone applications or libraries.

Web-based methods are not compared within this work as they most often require a complex infrastructure to be set up and maintained and are generally not available to external users for high data volumes. However, we will give a short overview of recent publications on this topic and provide short links to the parts of the metabolomics pipeline that we discuss here. A survey of web-based methods is provided by Tohge and Fernie (2009). More recent web-based applications for metabolomics include the retention time alignment methods Warp2D (Ahmad et al. 2011) and ChromA (Hoffmann and Stoye 2009), which are applicable to GC-MS or LC-MS data, and Chromaligner (S. Wang et al. 2010), which aligns GC and LC data with single-dimension detectors like FID.

Tools for statistical analysis of multiple sample groups and with different phenotypes have been reported by Kastenmüller et al. (2011). However, other tools aim to integrate a more complete metabolomics workflow including preprocessing, peakfinding, alignment and statistical analysis combined with pathway mapping information like MetaboAnalyst (Xia, Sinelnikov, and Wishart 2011), MetabolomeExpress (Carroll, Badger, and Millar 2010), or MeltDB (Kessler et al. 2013; Neuweger et al. 2008). These larger web-based frameworks integrate other functionality for time-course analysis (Xia, Sinelnikov, and Wishart 2011), pathway mapping (Xia and Wishart 2010; Neuweger et al. 2009) and metabolite set enrichment analysis (Kankainen et al. 2011; Xia and Wishart 2010).

We already defined metabolomics as the study of the metabolic state of an organism in response to direct or indirect perturbation. In order to find differences between two or more states, for example before treatment with a drug and after, and among

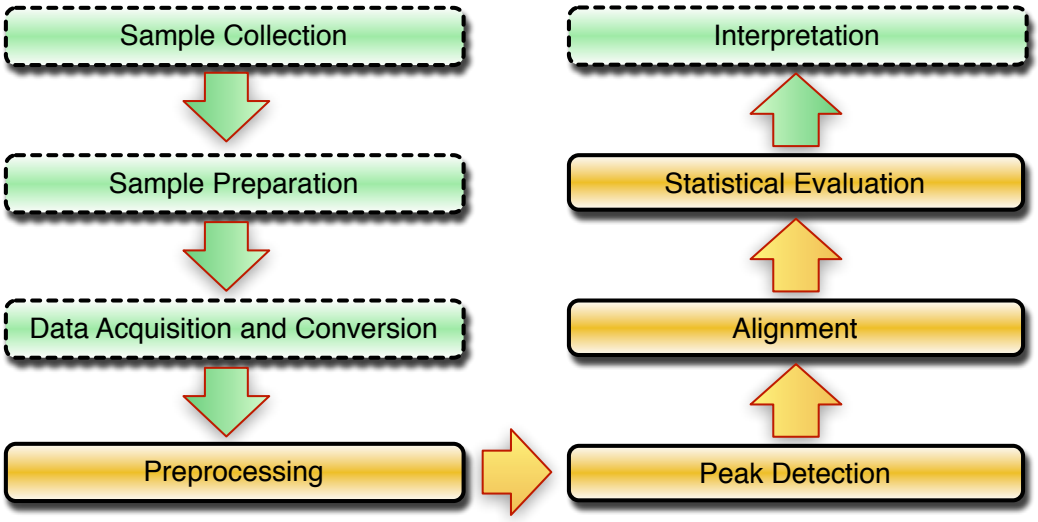


Figure 2.8.: A typical workflow for a metabolomics experiment. Steps shown in orange (solid border) are usually handled within the bioinformatics domain, while the steps shown in green (dashed border) often involve co-work with scientists from other disciplines.

one or multiple specimens, the actual hypothesis for the experiment needs to be defined. Based on this hypothesis, a design for the structure of the experiments and their subsequent analysis can be derived. This involves, among many necessary biological or medical considerations, the choice of sample extraction procedures and preparation methods, as well as the choice of the analytical methods used for downstream sample analysis.

Preprocessing of the data from those experiments begins after the samples have been acquired using the chosen analytical method, such as GC-MS or LC-MS. Owing to the increasing amount of data produced by high-throughput metabolomics experiments, with large sample numbers and high-accuracy/high-speed analytical devices, it is a key requirement that the resulting data is processed with a very high level of automation. It is then that the following typical workflow is applied in some variation, as illustrated in Figure 2.8.

2.6.1. Data Acquisition and Conversion

The most common formats exported from GC-MS and LC-MS machines today are NetCDF (Rew and Davis 1990), based on the specifications in the American Society for Testing and Materials (ASTM) standard ANDI-MS (Matthews and Miller 2000), mzXML (Pedrioli et al. 2004), mzData (Orchard et al. 2005), and more recently as the successor to the latter two, mzML (Martens et al. 2011; E. Deutsch 2008). All of these formats include well-defined data structures for meta-information necessary to

interpret data in the right context, such as detector type, chromatographic protocol, detector potential and other details about the separation and acquisition of the data. Furthermore, they explicitly model chromatograms and mass spectra, with varying degrees of detail.

NetCDF is the oldest and probably most widely used format today. It is routinely exported even by older machinery, which offers backwards compatibility to those. It is a general-purpose binary format, with a header that describes the structure of the data contained in the file, grouped into variables and indexed by dimensions. In recent years, efforts were made to establish open formats for data exchange based on a defined grammar in extensible markup language (XML)[3] with extendable controlled vocabularies, to allow new technologies to be easily incorporated into the file format without breaking backwards compatibility. Additionally, XML formats are human readable which narrows the technology gap. mzXML (Pedrioli et al. 2004) was the first approach to establish such a format. It was an alternative to the mzData format, (Orchard et al. 2004), with different approaches to modeling proteomics data in XML. More recently, mzML (Martens et al. 2011) was designed as a super-set of both, incorporating extensibility through the use of an indexed controlled vocabulary. This allows mzML to be adapted to technologies like GC×GC-MS without having to change its definition, although its origins are in the proteomics domain. Furthermore, mzML addresses the need for storing chromatographic data, for example the TIC, but also EICs. One drawback of XML-based formats is often claimed to be their considerably larger space requirements when compared to the supposedly more compact binary data representations. Recent advances in mzML approach this issue by compressing spectral data using gzip compression. An approach to overcome the encoding overhead was recently published as the mz5 format implementation (Wilhelm et al. 2011), which is based on the highly space efficient binary hierarchical data format (HDF5[4]), but maintains semantic compatibility to mzML, by using the same vocabulary and data model.

The data is continuously stored in a vendor-dependent native format during sample processing on a GC-MS or LC-MS machine. Along with the mass spectral information, like ion mass (or equivalents) and abundance, the acquisition time of each mass spectrum is recorded. Usually, the vendor software includes methods for data conversion into one of the aforementioned formats. However, especially when a high degreee of automation is desired, it may be beneficial to directly access the data in their native format. This avoids the need to run the vendor's proprietary software manually for every data conversion task. Both the ProteoWizard framework (Kessner et al. 2008) and the Trans Proteomic Pipeline (Deutsch et al. 2010) include multiple vendor-specific libraries for that use case.

3. http://www.w3.org/TR/REC-xml
4. http://www.hdfgroup.org

2.6.2. Preprocessing

Raw mass specrometry data is usually represented in sparse formats, only recording those masses whose intensities exceed a user-defined threshold. This thresholding is usually applied within the vendor's proprietary software and may lead to artificial *gaps* within the data. Thus, the first step in preprocessing involves the binning of mass spectra over time into bins of defined size in the m/z dimension, followed by interpolation of missing values. After binning, the data is stored as a rectangular array of values, with the first dimension representing time, the second dimension representing the approximate bin mass values, and the third dimension representing the intensity corresponding to each measured ion. This process is also often described as resampling (Lange et al. 2007).

Depending on various instrumental parameters, the raw exported data may require additional processing. The most commonly reported methods for smoothing are the Savitzky-Golay filter (Savitzky and Golay 1964), **LO**cal regr**ESS**ion (LOESS) (Smith et al. 2006) and variants of local averaging, for example by a windowed moving average filter. These methods can also be applied to interpolate values where gaps are present in the original data. The top-hat filter (Lange et al. 2007; Sturm et al. 2008) is used to remove a varying baseline from the signal. More refined methods use signal decomposition and reconstruction methods, such as Fourier transform and continuous wavelet transform (CWT) (Fredriksson et al. 2009; Tautenhahn, Böttcher, and Neumann 2008; Du et al. 2006) in order to remove noise and baseline contributions from the signal and simultaneously find peaks.

2.6.3. Peak Detection

Often the process of peak detection is decoupled from the actual preprocessing of the data. XCMS (Smith et al. 2006), for example, uses a Gaussian second derivative peak model with a fixed kernel width and signal-to-noise threshold to find peaks along the chromatographic domain of each ion bin. Other methods extend this approach to use a multi-scale continuous wavelet transform using such a kernel over various widths, tracking the response of the transformed signal in order to locate peak apex positions in scale-space before estimating the true peak widths based on the kernel scale with maximum response (Fredriksson et al. 2009; Tautenhahn, Böttcher, and Neumann 2008). However, these methods usually allow only a small number of co-eluting peaks in different mass-bins, since they were initially designed to work with LC-MS data mainly, where only one parent ion and a limited number of accompanying adduct ions are expected. In GC-MS, electron ionization creates rich fragmentation mass spectra that pose additional challenges to deconvolution of co-eluting ions, and to the subsequent association to peak groups. Even though its source code is not publicly available, the method used by AMDIS (Stein 1999) has seen wide practical application and is well accepted as a reference by the metabolomics and analytical chemistry communities. Two-dimensional chromatography provides additional challenges for peak detection, which are discussed for GC×GC-MS in Section 4.2.

2.6.4. Alignment

The alignment problem in metabolomics and proteomics stems from the analytical methods used. These produce sampled sensor readings acquired over time in fixed or programmed intervals, usually called chromatograms. The sensor readings can be one- or multidimensional. In the first case, detectors like ultra violet and visible light absorbance detectors (UV/VIS) or FIDs measure the signal response as one-dimensional features, e.g. as the absorbance spectrum or electrical potential, respectively. Multi-dimensional detectors like mass spectrometers record a large number of features simultaneously, e.g. mass and ion count. The task is then to find corresponding and non-corresponding features between different sample acquisitions. This *correspondence problem* is a term used by Åberg, Alm, and Torgrip (2009) which describes the actual purpose of alignment, namely to find *true* correspondences between related analytical signals over a number of sample acquisitions. For GC-MS and LC-MS-based data, a number of different methods have been developed, some of which are described in more detail by Castillo et al. (2011) and Åberg, Alm, and Torgrip (2009). Here, we will concentrate on those methods that have been reported to be applicable to GC-MS.

In principle, alignment algorithms can be classified into two main categories: peak- and signal-based methods. Methods of the first type start with a defined set of peaks, which are present in most or all samples that are to be aligned before determining the best correspondences of the peaks between samples in order to then derive a time correction function. Krebs et al. (2006) locate *landmark* peaks in the TIC and then select pairs of those peaks with a high correlation between their mass spectra in order to fit an interpolating spline between a reference chromatogram and the to-be-aligned one. The method of Robinson et al. (2007) is inspired by multiple sequence alignment algorithms and uses dynamic programming to progressively align peak lists without requiring an explicit reference chromatogram. Other methods, like that of Chae, Reis, and Thaden (2008) perform piecewise, block-oriented matching of peaks, either on the TIC, on selected masses, or on the complete mass spectra. Time correction is applied after the peak assignments between the reference chromatogram and the others have been calculated. Signal-based methods include recent variants of correlation optimized warping (COW) (Christin et al. 2008), parametric time warping (PTW) (Christin et al. 2010) and dynamic time warping (DTW) (Christin et al. 2010; Clifford et al. 2009; Hoffmann and Stoye 2009; Prince and Marcotte 2006) and usually consider the complete chromatogram for comparison. However, attempts are made to reduce the computational burden associated with a complete pairwise comparison of mass spectra by partitioning the chromatograms into similar regions (Hoffmann and Stoye 2009), or by selecting a representative subset of mass traces (Christin et al. 2010). Another distinction in alignment algorithms is the requirement of an explicit reference for alignment. Some methods apply clustering techniques to select one chromatogram that is most similar to all others (Christin et al. 2008; Hoffmann and Stoye 2009), while other methods choose such a reference based on the number of features contained in a chromatogram (Lange et al. 2007) or by manual user choice

(Clifford et al. 2009; Chae, Reis, and Thaden 2008). For high-throughput applications, alignments should be fast to calculate and reference selection should be automatic. Thus, a sampling method for time correction has recently been reported by Pluskal et al. (2010) for LC-MS. A comparison of these methods is given in the same publication.

The variability of retention times in both GC×GC and GC×GC-MS also requires sophisticated algorithms for automatic alignment of corresponding analyte signals between different samples. Additionally, the size and number of acquired sample data poses a significant challenge to automated methods and effectively prevents large scale manual intervention by human experts (Hoffmann and Stoye 2012). Due to the modulation of the signal, introduced by the second chromatographic column, peaks in GC×GC-MS are distributed over two retention time dimensions, with an additional non-linear shifting effect between signals of peaks that span multiple modulation periods.

The peak alignment problem for GC-MS data is addressed in Section 3.2. We also show that our method is applicable to the peak alignment problem for GC×GC-MS in Section 4.3, if a similarity function specific two the two-dimensional retention times is used.

2.6.5. Statistical Evaluation

After peaks have been located and integrated for all samples, and their correspondence has been established, peak report tables can be generated, containing peak information for each sample and peak, with associated corrected retention times and peak areas. Additionally, peaks may have been putatively identified by searching against a database, such as the Golm Metabolome Database (GMD) (Hummel et al. 2007) or the National Institute of Standards and Technology of the United States of America (NIST) mass-spectral database (Babushok et al. 2007). If mass spectral information is available, exact masses can be used to generate putative sum formulas (the elemental composition of an ion) and candidate structure formulas for ion peaks that are not contained in any public database (Neumann and Böcker 2010).

These peak tables can then be analyzed with further methods, in order to detect systematic differences between different sample groups. Prior to such an analysis, the peak areas need to be normalized. This is usually done by using a spiked-in compound as a reference that does not occur naturally in the studied organism. The normalization compound is supposed to have the same concentration in all samples. The compound's peak area can then be used to normalize all peak areas of a sample with respect to it (Doebbe et al. 2010) in relative units.

In order to associate these internally normalized peak areas to the original biological quantities, they can be normalized externally to their sample quantity of origin, e.g. the cell count, concentration, or dry weight of the sample. The peak areas are then represented in more biologically meaningful units.

Different experimental designs allow to analyze correlations of metabolite levels for the same subjects under different conditions (paired), or within and between groups of subjects. For simple paired settings, multiple t-tests with corrections for multiple

testing can be applied (Berk, Ebbels, and Montana 2011), while for comparisons between groups of subjects, Fisher's F-Statistic (Pierce et al. 2006) and various analysis of variance (ANOVA), principal components analysis (PCA) (Ventura et al. 2011) and partial least squares analysis (PLS) (Johnson et al. 2004) methods are applied (Kastenmüller et al. 2011; Xia, Sinelnikov, and Wishart 2011; Wiklund et al. 2008).

A more complete overview of current data processing methods and programs for GC×GC and GC×GC-MS data may be found in Matos, Duarte, and Duarte (2012) and Reichenbach et al. (2012), and Kallio et al. (2009).

2.6.6. Evaluation of Hypothesis

Finally, after peak areas have been normalized and differences have been found between sample groups, the actual results need to be put into their biological context. Many web-based analysis tools allow to interpret the results, by providing name- or id-based mapping of the experimentally determined metabolite concentrations onto biochemical pathways like MetaboAnalyst (Xia, Sinelnikov, and Wishart 2011), MetabolomeExpress (Carroll, Badger, and Millar 2010), or MeltDB (Neuweger et al. 2008; Kessler et al. 2013). The latter allows association of the metabolomics data with other results for the same subjects under study or with results from other *omics* experiments on the same target subjects that enables a more global, holistic approach called *systems biology* (Mesarović 1968), but this is beyond the scope of the frameworks and methods presented in this work.

3

Methods for GC-MS Data Analysis

In this chapter, we give an overview and feature comparison of existing Open Source frameworks for the handling and processing of data from GC-MS experiments. This overview covers the parts of the typical processing pipeline for metabolomics data that we introduced in Chapter 2.

We then explain the retention time alignment problem for multiple peak and profile data sets from GC-MS experiments in Section 3.2, explaining the novel methods **BI**directional best hits **P**eak **A**ssignment and **C**luster **E**xtension (BiPACE) (Section 3.3) and CeMAPP-DTW (Section 3.4) that we developed. Then we combine these two methods to create a new hybrid method that benefits from the speed and accuracy in peak matching of the peak-based alignment algorithm, while still providing a profile multiple alignment of all GC-MS datasets in reasonable time and space. We finally evaluate the algorithms against another state-of-the-art method in Section 3.5 and discuss the results in Section 3.6. BiPACE and CeMAPP-DTW were originally published in Hoffmann et al. (2012).

The algorithms presented in this chapter are available within our OpenSource framework **M**odular **A**pplication **T**oolkit for **C**hromatography-**M**ass Spectrometry (MaltCMS)[1]. We describe MaltCMS in more detail in Chapter 5.

3.1. Frameworks for GC-MS Analysis

A number of Open Source frameworks have been developed for LC-MS based proteomics frameworks like OpenMS (Sturm et al. 2008), ProteoWizard (Kessner et al. 2008), and most notably the TransProteomicPipeline (Deutsch et al. 2010). Even though many of the steps required for proteomics analysis apply similarly to metabolomics applications, there are still some essential differences due to the different analytical setups and technologies (e.g. matrix assisted laser desorption ionization mass spectrometry, MALDI-MS) used in the two fields. XCMS (Smith

1. http://maltcms.sourceforge.net

Table 3.1.: Overview of available Open Source software frameworks for GC-MS based metabolomics, their latest version, analytical methods covered, source-code and distribution license, as well as the main programming languages used in the framework. a: Part of Bioconductor 2.14, b: Eclipse Public License version 1.0.

Name	Version	Methods	License	Language
XCMS	1.39.4[a]	LC-MS/GC-MS	GPL v2	R >2.14, C++ 2003
PyMS	r375	GC-MS	GPL v2	Python 2.5
Maltcms	1.3	GC-MS/GC-FID	L-GPL v3, EPL v1[b]	Java 7
OpenChrom	0.8.0	GC-MS/GC-FID	EPL v1[b]	Java 7

et al. 2006) was among the first frameworks to offer support for data preprocessing in LC-MS based metabolomics. Later, MZmine2 (Pluskal et al. 2010) offered an alternative with a user-friendly interface and easy extensibility. Lately, Scheltema et al. (2011) published their PeakML format and mzMatch framework also for LC-MS applications. For an in-depths review of current LC-MS based metabolomics data preprocessing consider Castillo et al. (2011).

As of now, there are only a few frameworks available for GC-MS based metabolomics that offer similar methods, namely PyMS (Callaghan et al. 2010, 2012), Maltcms/ChromA (Hoffmann and Stoye 2009), and OpenChrom (Wenig and Odermatt 2010). These three, together with XCMS, will be presented in more detail in this section. A compact overview of the Open Source frameworks discussed herein is given in Table 3.1. A detailed feature comparison can be found in Table 3.2.

This overview excludes proprietary vendor software like Waters' MassLynx (Waters Corp., Milford, MA, USA), Agilent's ChemStation (Agilent Technologies, Inc., Santa Clara, CA, USA), Thermo's Xcalibur (Thermo Fisher Scientific Inc., Waltham, MA, USA), or LECO's ChromaTOF (LECO Corp., St. Joseph, MI, USA) for a number of reasons.

The first reason are the limited capabilities of proprietary software concerning data interoperability. Usually, only netCDF following either the ANDI-MS or ANDI-CHROM (Matthews and Miller 2000) standards is supported. Sometimes, support for an additional open format, such as mzXML (Pedrioli et al. 2004), mzData[2], or lately, mzML (E. Deutsch 2008) is offered. However, the exported files do not always follow standard conventions or add custom, non-standardized information, which makes it hard to read these files by other, downstream software. The interoperability aspect has improved lately by the strict validation requirements introduced by the mzML data standard, and today most major companies working in the field offer at least one method to access the raw data. The gaps that the vendor softwares leave open are addressed by growing companies like GeneData (Basel, Switzerland), with

2. http://psidev.info/mzdata

their Expressionist software, that provide direct access to many proprietary vendor data formats.

The second reason against proprietary software is the non-availability of the implementation of the algorithms used for data processing, peak finding and integration, and multiple alignment. Usually, only a very shallow description of the algorithm is given in the user manual of proprietary software, with often insufficient explanations about the influence of individual parameters.

The third reason is the platform dependence of the proprietary software. Current vendor software operates mainly under Microsoft Windows operating systems, while the Open Source frameworks described in the following section are all platform independent and can be run on a large variety of different hardware and operating systems, giving researchers more flexibility, also in terms of using computer grid resources for large scale data processing tasks without the requirement for a graphical user interface.

3.1.1. XCMS

XCMS (Smith et al. 2006) is a very mature framework and has seen constant development during the last five years. It is mainly designed for LC-MS applications, however its binning, peak finding and alignment are also applicable to GC-MS data. XCMS is implemented in the R[3] programming language, the de-facto standard for Open Source statistics. Since R is an interpreted scripting language, it is easy to write custom scripts that realize additional functionality of the typical GC-MS workflow described above. XCMS is part of the Bioconductor[4] package collection, which offers many computational methods for various "omics" technologies. Further statistical methods are available from R and auxiliary packages.

XCMS supports input in NetCDF, mzXML, mzData and, more recently, mzML format. This allows XCMS to be used with virtually any chromatography-mass spectrometry data, since vendor software supports conversion to at least one of those formats. XCMS uses the *xcmsRaw* object as its primary tabular data structure for each binned data file. The *xcmsSet* object is then used to represent peaks and peak groups and is used by its peak alignment and *diffreport* features.

The peak finding methods in XCMS are quite different from each other. For data with normal or low mass resolution and accuracy, the matched filter peak finder (Smith et al. 2006) is usually sensitive enough. It uses a Gaussian peak template function with user defined width and signal-to-noise criteria to locate peaks on individual binned EIC traces over the complete time range of the binned chromatogram. The other method, CentWave (Tautenhahn, Böttcher, and Neumann 2008) is based on a continuous wavelet transform on areas of interest within the raw data matrix. Both peak finding methods report peak boundaries and integrated areas for raw data and for the data reconstructed from the peak finder's signal response values.

3. http://www.r-project.org
4. http://www.bioconductor.org

Table 3.2.: Feature comparison of Open Source software frameworks for preprocessing of GC-MS based metabolomics data. Keys to abbreviations: **Data formats** A: NetCDF, B: mzXML, C: mzData, D: mzML, E: JCAMP GC-MS, F: Proprietary. **Signal preprocessing** MM: moving median, SG: Savitzky-Golay filter, TH: top-hat filter, MA: moving average. **Peak detection** MF: matched Gaussian filter, CWT: continuous wavelet transform, BB: Biller-Biemann, MAX: TIC local maxima. **Multiple peak alignment** LOESS: LOESS regression, DTW: dynamic time warping, PROGDP: progressive using dynamic programming, CLIQUE: progressive clique-based. **Visualization** (of unaligned and aligned data) TIC: plots of total ion chromatogram/peaks, EIC: plots of extracted ion chromatograms/peaks, SURF: surface plots of profile matrix (rt ×m/z ×I). **DB search** MSP: msp-format, compatible with AMDIS and GMD format, OWN: proprietary format. **Normalization** RP: reference peak area, EV: external value, e.g. dry weight. **Statistical evaluation** TT: groupwise t-test, multiple testing correction, FT: F-test, between group vs. within group variance, AOV: analysis of variance, PCA: principal components analysis. a: via Rserve in MAUI.

Feature (GC-MS pipeline)	XCMS	PYMS	MALTCMS/CHROMA	OPENCHROM
Data formats	A, B, C, D	A, E	A, B, C, D	A, B, C, D, F
Signal preprocessing	MM	SG, TH	SG, MA, MM, TH	MA, SG
Peak detection / integration	MF, CWT	BB	MAX, CWT	MAX
Multiple peak alignment	LOESS, DTW	PROGDP	DTW, CLIQUE	–
Visualization	TIC, EIC, SURF	TIC, EIC	TIC, EIC, SURF	TIC, EIC, SURF
DB search	– (LC-MS only)	–	MSP	MSP, OWN
Normalization	–	–	RP, EV	RP, EV
Statistical evaluation	TT	–	FT, (AOV, PCA)[a]	–

Initially designed for LC-MS, XCMS does not have a method to group co-eluting peaks into peak groups, as is a requirement in GC-MS methods using electron ionization. However, CAMERA (Tautenhahn, Böttcher, and Neumann 2007) and XCMS Online (Tautenhahn et al. 2012) show how XCMS can be used as a basis in order to create a derived application for ion annotation between samples and untargeted metabolomics, respectively.

Peak alignment in XCMS is performed using LOESS regression between peak groups with very similar m/z and retention time (RT) behavior and good support within each sample group. This allows a simultaneous alignment and retention time correction of all peaks. The other available method is based on the Obi-Warp DTW algorithm (Prince and Marcotte 2006) and is capable of correcting large non-linear RT distortions. It uses the peak set with the highest number of features as alignment reference, which is comparable to the approach used by Lange et al. (2007). However, it is much more computationally demanding then the LOESS-based alignment.

XCMS's *diffreport* generates a summary report of significant analyte differences between two sample sets. It uses Welch's two-sample t-statistic to calculate probability values (p-values) for each analyte group. ANOVA may be used for more than two sample sets.

A number of different visualizations are also available, for both raw and processed data. These include TIC plots, EIC plots, analyte group plots for grouped features, and chromatogram (RT, m/z, intensity) surface plots.

XCMS can use GNU R's Rmpi infrastructure to execute arbitrary function calls, such as profile generation and peak finding, in parallel on a local cluster of computers.

3.1.2. PyMS

PyMS (Callaghan et al. 2010, 2012) is a programming framework for GC-MS metabolomics based on the Python programming language[5]. It can therefor use many scientific libraries which are accessible via the SciPy and NumPy packages[6]. Since Python is a scripting language, it allows to do rapid prototyping, comparable to GNU R. However, Python's syntax may be more familiar for programmers with a background in object-oriented programming languages.

The downloadable version of PyMS currently only supports NetCDF among the more recent open data exchange formats. Nonetheless, it is the only framework in this comparison with support for the JCAMP GC-MS file format.

PyMS provides dedicated data structures for chromatograms, allowing efficient access to EICs, mass spectra, and peak data.

In order to find peaks, PyMS also builds a rectangular profile matrix with the dimensions time, m/z and intensity. Through the use of slightly shifted binning boundaries, they reduce the chance of false assignments of ion signals to neighboring bins, when binning is performed with unit precision (bin width of 1 m/z). PyMS

5. http://www.python.org
6. http://www.scipy.org

offers the moving average and the Savitzky-Golay filters (Savitzky and Golay 1964) for signal smoothing of EICs within the profile matrix. Baseline correction can be performed by the top-hat filter (Lange et al. 2007). The actual peak finding is based on the method described by Biller and Biemann (1974) and involves the matching of local peak maxima co-eluting within a defined window. Peaks are integrated for all co-eluting masses, starting from a peak apex to both sides and ending if the increase in area falls below a given threshold.

Peak alignment in PyMS is realized by the method introduced by Robinson et al. (2007). It is related to progressive multiple sequence alignment methods and is based on a generic dynamic programming algorithm for peak lists. It proceeds by first aligning peak lists within sample groups, before aligning the aligned peak lists of different groups, until all groups have been aligned.

Visualizations of chromatogram TICs, EICs, peaks and mass spectra are available and are displayed to the user in an interactive plot panel.

For high-throughput applications, PyMS can be used together with MPI to parallelize tasks within a local cluster of computers.

3.1.3. OpenChrom

OpenChrom (Wenig and Odermatt 2010) offers a convenient graphical user interface for GC-MS and gas chromatography-flame ionization detector (GC-FID) data within the area of analytical chemistry. It is implemented in the JAVA programming language, based on the Eclipse Rich Client Platform module infrastructure[7].

OpenChrom provides direct support for most proprietary vendor formats, peak finding and both automatic and manual peak integration, as well as custom mass spectral database creation and analyte identification. It uses a custom binary format to store its own chromatogram data .

Filtering of chromatograms (TIC, EIC) prior to peak finding can be performed with the Savitzky-Golay filter (Savitzky and Golay 1964) and the component detection algorithm (CODA) (Windig, Phalp, and Payne 1996). OpenChrom's peak finder and integrator work comparably to the method used in Agilent's ChemStation (Agilent, Santa Clara CA, USA), using first and second derivatives to determine peak maxima, minima, and inflection points, generating a parameterized function for each peak.

Retention time correction can be performed individually for each peak. As of version 0.8.0, there is no support for an automatic retention time correction. However, if peaks have been identified, they can be used to create a compound-abundance matrix for all measured samples and annotated peaks.

OpenChrom provides many interactive views to browse and visualize chromatographic and peak data, also allowing for manual peak integration and annotation. It also allows to define configurations for linear batch processing of chromatograms using the same methods that are available within the user interface.

7. http://eclipse.org

3.1.4. Maltcms

The framework MALTCMS[8] allows to set up and configure individual processing components for various types of computational analyses of metabolomics data. The framework is implemented using the JAVA programming language[9] and is modular using the service provider pattern for maximal decoupling of interface and implementation, so that it can be extended in functionality at runtime.

MALTCMS can read data from files in NetCDF, mzXML, mzData or mzML format. It uses a pipeline paradigm to model the typical preprocessing workflow in metabolomics, where each processing step can define dependencies on previous steps. This allows automatic pipeline validation and ensures that a user can not define an invalid pipeline. The workflow itself is serialized to XML format, keeping track of all resources created during pipeline execution. Using a custom post-processor, users can define which results of the pipeline should be archived.

MALTCMS uses a generalization of the ANDI-MS data schema internally and a data provider interface with corresponding implementations to perform the mapping from any proprietary data format to an internal data object model. This allows efficient access to individual mass spectra and other data available in the raw-data files. Additionally, developers need no special knowledge of any supported file format, since all data can be accessed generically. Results from previous processing steps are referenced in the data model to allow both shadowing of data, e.g. creating a processing result variable with the same name as an already existing variable, and aggregation of processing results. Thus, all previous processing results are transparently accessible for downstream elements of a processing pipeline, unless they have been shadowed.

Primary storage of processing results is performed on a per-chromatogram basis in the binary NetCDF file format. Since metabolomics experiments create large amounts of data, a focus is put on efficient data structures, data access, and scalability of the framework.

Embedding MALTCMS in existing workflows or interfacing with other software is also possible, as alignments, peak-lists and other feature data can be exported as comma separated value files or in specific XML-based formats, which are well-defined by custom schemas.

To exploit the potential of modern multi-core CPUs and distributed computing networks, MALTCMS supports multi-threaded execution on a local machine or within a grid of connected computers using an OpenGrid infrastructure (e.g. Oracle Grid Engine or Globus Toolkit (Foster 2005)) or a manually connected network of machines via remote method invocation (RMI[10]). More details on the parallel execution framework can be found in Chapter 5.

8. http://maltcms.sf.net
9. http://www.oracle.com/technetwork/java/javase/overview
10. http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html

The framework is accompanied by many libraries for different purposes, such as the *JFreeChart*[11] library for 2D-plotting or, for BLAS compatible linear algebra, math and statistics implementations, the *Colt*[12] and *commons-math*[13] libraries. Building upon the base library CROSS[14], which defines the commonly available interfaces and default implementations, MALTCMS provides the domain dependent data structures and specializations for processing of chromatographic data.

ChromA

Chromatogram **A**lignment (CHROMA) is a configuration of MALTCMS that includes preprocessing, in the form of mass binning, time-scale alignment and annotation of signal peaks found within the data, as well as visualizations of unaligned and aligned data from GC-MS and LC-MS experiments. The user may supply mandatory alignment anchors as comma separated value (CSV) files to the pipeline and a database location for tentative metabolite identification. Further downstream processing can be performed either on the retention time-corrected chromatograms in NetCDF format, or on the corresponding peak tables in either CSV format or XML format.

Peaks can be imported from other tools by providing them in CSV format to CHROMA, requiring at least the scan index of each peak in a file per row. Alternatively, CHROMA has a fast peak finder that locates peaks based on derivatives of the smoothed and baseline-corrected TIC, using user-definable signal-filters and a LOESS-based baseline estimation, with a customizable minimum peak-to-peak-apex window. Peak alignment is based on a star-wise or tree-based application of an enhanced variant of pairwise DTW (Hoffmann and Stoye 2009). To reduce both runtime and space requirements, conserved signals throughout the data are identified, constraining the search space of DTW to a precomputed closed polygon. The alignment anchors can be augmented or overwritten by user-defined anchors, such as previously identified compounds, characteristic mass or MS/MS identifications. Then, the candidates are paired by means of a bidirectional best-hit (BBH) criterion, which can compare different aspects of the candidates for similarity. Paired anchors are extended to k-cliques with configurable k, which help to determine the conservation or absence of signals across measurements, especially with respect to replicate groups. Tentative identification of peaks against a database using their mass spectra is possible using the MetaboliteDB module. This module provides access to mass-spectral databases in MSP-compatible format, for example the Golm Metabolite Database or the NIST EI-MS database.

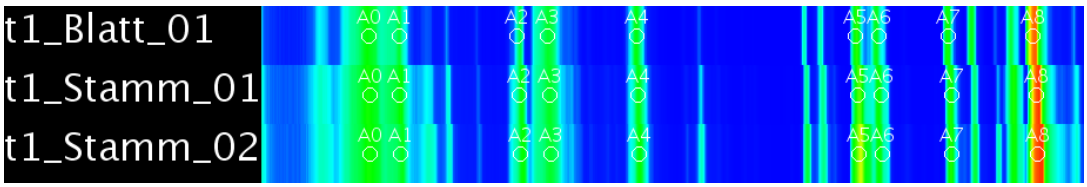
CHROMA visualizes alignment results including paired anchors in birds-eye view or as a simultaneous overlay plot of the TIC. Additionally, absolute and relative differential charts are provided, which allow easy spotting of quantitative differences.

11. http://www.jfree.org/jfreechart
12. http://acs.lbl.gov/software/colt
13. http://commons.apache.org/proper/commons-math
14. http://sf.net/p/maltcmscross

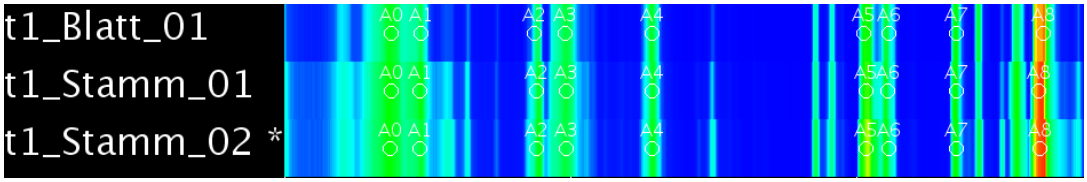
Peak tables are exported in CSV format, including peak apex positions, area under curve, peak intensity and possibly tentative database identifications. Additionally, information about the matched and aligned peak groups is saved in CSV format.

3.2. Multiple Alignment of GC-MS Chromatograms

Metabolomics, the study of an organism's biochemistry, has become increasingly relevant along with other *omics* technologies during the last ten years. Some of the techniques of choice to distinguish the metabolites present in a biological sample of an organism are separation techniques coupled to sensitive detectors, such as GC-MS and LC-MS. In contrast to FIDs, UV absorbance detector (UVD), and other one-dimensional detectors, these hyphenated methods provide high-dimensional data of analyte molecular ions or analyte molecular ion fragments collected over the runtime of the separation. In the context of metabolomics, this usually involves the observation of potentially hundreds of ion signals of different masses simultaneously in every recorded scan. These numbers may be even higher for proteomics, owing to the larger masses of peptides and peptide fragments. Comparing such data manually to find corresponding signals is very labor intensive, as each experiment usually consists of thousands of individual scans. Thus, the goal must be to obtain a high level of automation during data acquisition and data processing, allowing scientists to focus on the informative parts of their data, while still alerting them to potential errors or problems.



(a) TIC view of unaligned chromatograms.



(b) TIC view of chromatograms aligned with BIPACE and CEMAPP-DTW .

Figure 3.1.: TIC view sections of unaligned and aligned chromatograms with highlighted alignment anchors, as determined by BIPACE.

Often it is the goal of a metabolomics experiment to detect differences between a treated and a control group of measurements. Therefore, an accurate alignment and matching of corresponding features in all measurements is an extremely important part of data preprocessing. Data matrices representing the detected and aligned

features across all measurements may be generated in order to be used for further statistical analysis. It is essential that an alignment algorithm captures fluctuations in the chromatographic system that lead to non-linear distortions of the retention time of individual features (Podwojski et al. 2009; Strehmel et al. 2008). Further, it needs to group those features that are most similar to each other and to discover whether features are present or absent. An example of three chromatograms showing retention time variation within the first few minutes of the chromatography before and after alignment is shown in Figure 3.1.

In the end, a matrix of grouped peak features of single or related coeluting analyte ions should be generated to establish relationships in abundance between different experimental conditions. Then, based on other characteristics such as parent ion mass, ion fragments or isotope pattern, an identification of those features for integration with downstream analysis is required. Here we focus on the first few steps of such an analysis pipeline, including the generation of a matrix of grouped features for retention time normalization.

The currently available algorithms for retention time alignment can be distinguished into two general categories: peak-based and raw data-based alignment. The peak-based algorithms require prior peak or feature-finding and often also peak deconvolution to reduce the effect of overlapping signals, before a score function is applied to establish correspondence between peaks (Chae, Reis, and Thaden 2008; Styczynski et al. 2007; Lange et al. 2007; Krebs et al. 2006; Smith et al. 2006). Raw data-based algorithms on the other hand require little or no preprocessing, but are computationally very expensive (Prince and Marcotte 2006; Prakash et al. 2006).

3.2.1. Peak-based algorithms

Peak-based algorithms are very sensitive to the correctness of the a priori peak detection. A peak may be defined as the time-resolved signal intensity trace of an analyte ion's corresponding mass matching predefined criteria, such as the goodness-of-fit to a predefined peak model shape, together with a signal-to-noise ratio threshold (Smith et al. 2006). If a peak is tagged to be absent during preprocessing, it cannot be aligned by a peak-based algorithm. In order to handle missing peaks in data matrices for statistical analysis, Smith et al. then fill the gaps by using estimates based on prior grouping of the data. Such a grouping usually consists of at least two groups, e.g. control and treated group. Then, for a peak missing within a group, where most other peaks are present, the missing value can be estimated from the present members of the group. However, such peak imputation may be erroneous if it is only based on the final peak tables and does not access the original data to ensure that a peak is really present.

To be able to assign peaks that may not have been aligned, Krebs et al. (2006) proposed an approach based on prior peak detection and grouping, followed by polynomial interpolation to infer warping in between grouped peaks. Prince and Marcotte

(2006) introduced a similar interpolation scheme for raw data-based alignment with dynamic time warping.

A further division of peak-based algorithms may also be applied concerning the use of mass spectra for peak similarity calculation. Warping based on peaks detected in the TIC is usually supplemented by using MS, to increase the number of true positive peak assignments (Styczynski et al. 2007; Robinson et al. 2007; Krebs et al. 2006). Some algorithms work on a more complete set of extracted features, e.g. points of retention time, m/z and intensity (Lange et al. 2008; Jaitly et al. 2006), but often resort to linear regression in order to compute a retention time correction, due to the large amount of points that need to be processed. A more exhaustive overview of existing feature-based alignment algorithms to align point sets is given by Lange et al. (2008), especially for the application to LC-MS data in proteomics and metabolomics. Åberg, Alm, and Torgrip (2009) described the peak correspondence problem for NMR, showing that there is a significant amount of overlap considering the algorithms for these, at first sight different, application domains.

3.2.2. Raw data-based algorithms

Raw data-based algorithms operate on the complete collection of (binned) MS data, also termed the *uniform matrix*, such as ObiWarp (Prince and Marcotte 2006), which is based on DTW between binned mass spectra using pairwise spectra similarities, or the signal maps approach by Prakash et al. (2006). Therefore, these algorithms should find more and possibly better correspondences compared to the peak-based algorithms, which only have access to a limited number of reported peak features. Other approaches use COW (Bylund et al. 2002) for TIC alignment, or generalizations thereof (Christin et al. 2010; Ramaker et al. 2003), selecting specific mass traces to improve over simple TIC-based alignment. However, using many mass traces increases the computational demand, as well as the amount of data in need of processing, and may also increase the tendency of aligning noise (Windig, Phalp, and Payne 1996; Christin et al. 2010). Possibly owing to that computational demand, most raw data-based algorithms do not consider alignment or matching of individual points of retention time, m/z and intensity, but instead only try to correct the retention time deviation for each mass spectrum as a whole. The advantage of raw data-based methods is that they assign a definite position to each mass spectrum together with its corrected retention time after alignment. They use a pairwise similarity function between either TIC or sequences of mass spectra, finding an optimal global similarity with respect to their objective function (Clifford et al. 2009; Pierce, Wright, and Synovec 2007; Eilers 2004). The local correspondences between two raw data sets then allow to select the mass spectra with the highest pairwise similarities after the alignment to pinpoint peaks of interest for further investigation (Prince and Marcotte 2006).

3.2.3. Structure

In the next sections we introduce two novel methods for retention time alignment of multiple GC-MS and LC-MS experiments, which may be used individually and in combination as a hybrid method. The first method, BIPACE, is related to the clique-finding method described by Styczynski et al. (2007), but without relying on deconvoluted peaks and choosing a different criterion for peak correspondence and clique coherence, which drastically decreases computation times. It is a peak-based alignment method that automatically finds conserved groups of peaks among an arbitrary collection of chromatograms, based on the bidirectional best hit criterion as introduced by Tatusov, Koonin, and Lipman (1997) and later by Overbeek et al. (1999) for the matching of orthologous genes. Peaks are compared using user-definable similarities based on their mass spectra, for example with the similarity introduced by Robinson et al. (2007), or by derived similarity functions, that we will introduce in this work, and are successively grouped into clusters of best pairwise correspondence. This method allows to find clusters of arbitrary size, up to the number of chromatograms under consideration. It may be applied to different experimental protocols with more than just two groups of treatment and control, since the algorithm requires no prior knowledge of an existing grouping.

The second method, CEMAPP-DTW, applies DTW as in (Prince and Marcotte 2006), but to all pairs of chromatograms. DTW was first introduced and used in speech recognition for the alignment of time dependent feature traces of speech samples (Itakura 1975; Sakoe and Chiba 1978; Kruskal and Liberman 1983). One of the first applications of alignment methods to low-resolution GC-MS data was performed by Reiner et al. (1979), based on the local squared distance of the TIC. More recent applications have been reported by Christin et al. (2010), Clifford et al. (2009), Prince and Marcotte (2006), and Ramaker et al. (2003). Prince and Marcotte showed that different local score or cost functions can be used in order to align data from LC-MS experiments with good performance. Other methods for the alignment of raw chromatographic data exist, such as aligning the time series data to a latent trace, which is constructed from training series, with an underlying stochastic model (Listgarten et al. 2005) or by different means of regression (Fischer, Roth, and Buhmann 2007). We use the grouped peaks from BIPACE as anchors to constrain the pairwise DTW alignments, as outlined in a previous publication (Hoffmann and Stoye 2009). This results in faster computation and at the same time considerably less memory usage than in the unconstrained cases through the use of an optimized data structure, while providing comparable alignment results. Building on the pairwise alignments, we choose the chromatogram with the highest sum of pairwise similarities as the reference for the final alignment of all remaining chromatograms to the reference. We use DTW to compute the pairwise alignment, due to its applicability to data with non-linear time scale distortions, its relatedness to classical sequence alignment algorithms (Itakura 1975; Sakoe and Chiba 1978; Kruskal and Liberman 1983) and its proven power to perform retention time correction and signal alignment (Christin et al. 2010; Prince and Marcotte 2006; Ramaker et al. 2003).

3.3. BIPACE

Given a chromatogram $C = \{p_1, p_2, ..., p_\ell\}$ as an ordered set of ℓ peaks, we define a peak $p = (\mathbf{m}, \mathbf{i}, t)$ as a triple of a mass vector $\mathbf{m}$, an intensity vector $\mathbf{i}$, both of length N, and a retention time t. Peaks can be matched between chromatograms by exhaustive search, if a feasible criterion for their similarity exists. Based on GC-MS EI fragmentation mass spectra alone, such a criterion is hard or even impossible to find especially due to the ambiguity of the mass spectra of isomers. Additionally, we have to deal with inherent noise, introduced by contaminations of the sample from external sources (sample preparation) or internal sources (sample injection, chromatographic system, MS acquisition). Thus, we use a similarity function $s(p, q)$ between peaks p and q, represented as (nominal) mass intensity vectors, like the cosine similarity (see Definition 3.3) weighted by an exponentially penalized difference in RT (acquisition time) of the spectra (Robinson et al. 2007). For two peaks $p = (\mathbf{m}_p, \mathbf{i}_p, t_p)$ and $q = (\mathbf{m}_q, \mathbf{i}_q, t_q)$ and a retention time tolerance of D, we define this similarity function as follows:

Definition 1 (BIPACE RT Similarity Function)**.**

$$f(p, q) := \exp\left(-\frac{(t_p - t_q)^2}{2D^2}\right) \cdot s(p, q) \ . \tag{3.1}$$

The effect of the Gaussian RT difference function is that of a weighting function. Thus, for perfect RT correspondence between two peaks, the weight will be 1, giving full weight to the value of the mass spectral similarity. If the RTs of the two peaks differ, the weight will quickly decrease towards 0, depending on D. The decrease will be slower for high values of D (large retention time deviation), while it will be fast for low values of D (small retention time deviation). The similarity function s would typically be the cosine value (Equation 3.3) of the angle between the two peaks' mass spectral intensity vectors: $s(p, q) = \cos \angle(\mathbf{i}_p, \mathbf{i}_q)$. However, s can also be realized by any other similarity function defined between two vectors, such as the negative Euclidean distance, the dot product, Pearson's linear correlation or Spearman's rank correlation. We will now define these pairwise mass spectral similarities based on our notation.

Definition 2 (Dot Product)**.**

$$s(p, q) := \mathbf{i}_p \cdot \mathbf{i}_q = \sum_{j=1}^{N} \mathbf{i}_{p,j} \mathbf{i}_{q,j} \ . \tag{3.2}$$

In terms of our definition above, the dot product between two mass spectral intensity vectors is defined as the sum over all component-wise products of the vectors. The dot product takes on values between 0, indicating orthogonality, and $+\infty$. It is maximal for identical vectors, but in principle unbound.

Definition 3 (Cosine Similarity.)**.**

$$s(p,q) := \cos\angle(\mathbf{i}_p, \mathbf{i}_q) = \frac{\mathbf{i}_p \cdot \mathbf{i}_q}{\|\mathbf{i}_p\| \, \|\mathbf{i}_q\|} = \frac{\sum_{j=1}^{N} \mathbf{i}_{p,j}\mathbf{i}_{q,j}}{\sqrt{\sum_{j=1}^{N} \mathbf{i}_{p,j}^2}\sqrt{\sum_{j=1}^{N} \mathbf{i}_{q,j}^2}} \,. \tag{3.3}$$

The cosine value of the angle between two mass spectral intensity vectors is based on the dot product, divided by the product of the Euclidean norms of each vector individually. The value of the cosine score lies in the closed interval $[-1, 1]$, but can be restricted to lie within $[0, 1]$ for non-negative intensity vectors. Co–linear vectors have a cosine similarity of 1, regardless of scaling, while the similarity is 0 for orthogonal vectors. The cosine and the dot product similarity are related via the following identity: $\mathbf{i_p} \cdot \mathbf{i_q} = \|\mathbf{i_p}\| \, \|\mathbf{i_q}\| \cos\angle(\mathbf{i}_p, \mathbf{i}_q)$.

Definition 4 (Pearson's Linear Correlation)**.**

$$s(p,q) := \frac{\sum_{j=1}^{N} \left(\mathbf{i}_{p,j} - \bar{\mathbf{i}}_p\right)\left(\mathbf{i}_{q,j} - \bar{\mathbf{i}}_q\right)}{\sqrt{\sum_{j=1}^{N} \left(\mathbf{i}_{p,j} - \bar{\mathbf{i}}_p\right)^2}\sqrt{\sum_{j=1}^{N} \left(\mathbf{i}_{q,j} - \bar{\mathbf{i}}_q\right)^2}} \,. \tag{3.4}$$

The value of the linear correlation coefficient is in the closed interval $[-1, 1]$, where -1 indicates perfect anti-correlation, meaning that if p is high then q is low, and vice versa. A value of 0 is attained if p and q are not linearly correlated. However, this still allows higher order correlations of p and q that are not immediately measurable with Pearson's method. A value of 1 indicates perfect correlation, meaning that p and q differ at most by a constant linear factor in all dimensions. The similarity is invariant to linear scaling so that no prior length normalization of the intensities is necessary. The same applies to the cosine similarity (Eqn. 3.3).

Definition 5 (Spearman's Rank Correlation)**.**

$$s(p,q) := \frac{\sum_{j=1}^{N} \left(rk(\mathbf{i}_{p,j}) - \overline{rk_{\mathbf{i}p}}\right)\left(rk(\mathbf{i}_{q,j}) - \overline{rk_{\mathbf{i}q}}\right)}{\sqrt{\sum_{j=1}^{N} \left(rk(\mathbf{i}_{p,j}) - \overline{rk_{\mathbf{i}p}}\right)^2}\sqrt{\sum_{j=1}^{N} \left(rk(\mathbf{i}_{q,j}) - \overline{rk_{\mathbf{i}q}}\right)^2}} \,, \tag{3.5}$$

where $\mathrm{rk}(\mathbf{i}_{p,j})$ is the rank of intensity value j within p, and $\overline{\mathrm{rk}_{\mathbf{i}p}}$ is the average rank of all intensities in $\mathbf{i}_p$. This holds likewise for $\mathbf{i}_q$. The rank correlation ρ is a more robust variant of the linear correlation, especially if outliers can be expected in the data. In case of the linear correlation, these would severely degrade the correlation value, even though their omission would lead to an almost perfect correlation value. By focusing on the ranks of the data, rather than the absolute values, the rank correlation is less sensitive to such outliers, as they will at most influence a small number of ranks. For mass spectral comparison, the rank correlation seems to be most adequate for faint and weak signals, that also exhibit a large variation between measurements. In most other cases, the linear correlation and cosine similarity will outperform it (see Section 3.5 for more details).

Definition 6 (Negative Euclidean Distance)**.**

$$s(p,q) := -\sqrt{\sum_{j=1}^{N} \left(\mathbf{i}_{p,j} - \mathbf{i}_{q,j}\right)^2}\,. \tag{3.6}$$

The Euclidean distance sums the squared differences in values for each feature dimension, in this case the intensities of corresponding mass bins between peaks p and q. It is very susceptible to noise in the data and as such is hard to optimize for a large range of different datasets. It may work better if the log intensities are used instead of the raw ones, but this has not been tested in this work. However, it is the only similarity that has metric properties: it is non-negative, $s(p,q) = 0$ if and only if $p = q$, it is symmetric: $s(p,q) = s(q,p)$, and it fulfills the triangle inequality: $s(p,q) \leq s(p,r) + s(r,q)$ for any third peak r.

Definition 7 (Weighted Cosine Similarity)**.**

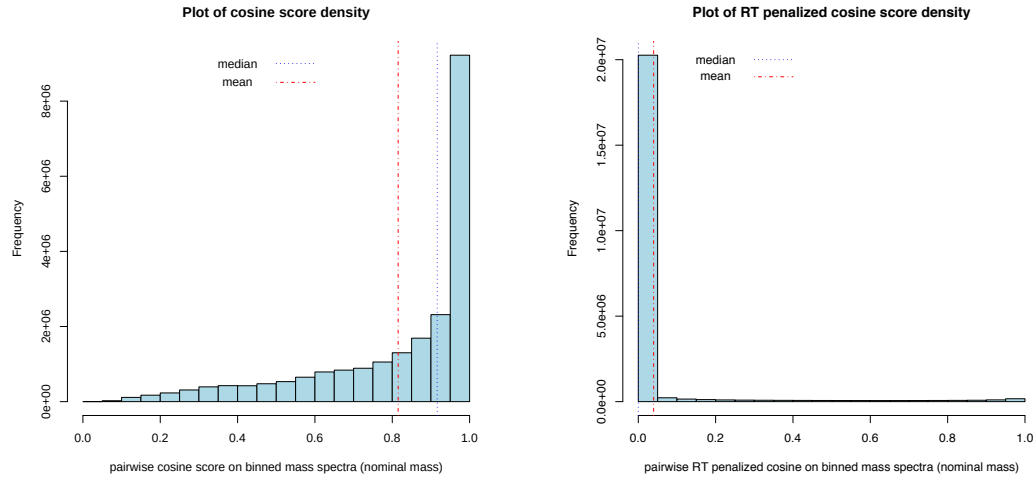
$$s(p,q) := \frac{\sum_{j=1}^{N} \left((\mathbf{m}_{p,j}\mathbf{m}_{q,j})^u (\mathbf{i}_{p,j}\mathbf{i}_{q,j})^v\right)^2}{\left(\sum_{j=1}^{N} \mathbf{m}_{p,j}^{2u}\mathbf{i}_{p,j}^{2v}\right)\left(\sum_{j=1}^{N} \mathbf{m}_{q,j}^{2u}\mathbf{i}_{q,j}^{2v}\right)}\,. \tag{3.7}$$

The weighted cosine similarity was first introduced by Stein and Scott (1994). For each peak compared, it requires prior normalization of the intensities to the maximum of the peak's intensities. The parameters u and v control the individual influence of mass and intensity terms on the overall score. Typically, u is set to 1, while v is set to 0.5, effectively giving higher masses a higher weight (Castillo et al. 2011).

Essentially, all similarity functions that were defined above on the domain $[-1,1]$ will only attain values in the interval $[0,1]$ on the positive intensity values that are typically present in mass spectra.

The similarity function f in combination with any of the similarity functions just presented leads to a good pre-filtering of candidate peaks for matching throughout our input chromatograms. The effect of combining the cosine similarity with the retention time penalty on the score distribution is shown in Figure 3.2 for two related and typical GC-MS datasets using an EI detector. Without retention time penalization, the score distribution is heavily biased towards a median value slightly above 0.9, meaning that more than 50% of the pairwise similarities have a value > 0.9. However, together with the retention time penalty function, this is reduced, so that the 50%-tile of the similarities is below a value of 0.05. The number of peaks that have similarities > 0.9 is now greatly decreased and illustrates the filtering effect that the Gaussian retention time penalty has.

Pairwise Similarity Calculation In order to assign peaks to their best corresponding counterparts, we calculate all pairwise similarities using the similarity function f between all peaks from distinct chromatograms. The similarity calculation is



(a) Cosine score distribution for a typical GC-MS / EI dataset.

(b) Time penalized cosine score distribution for a typical GC-MS / EI dataset.

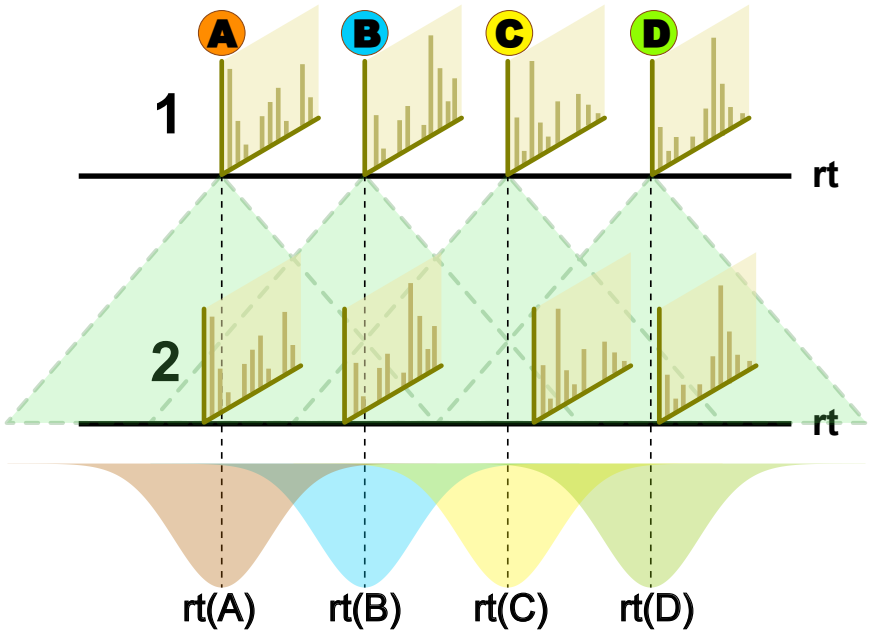
Figure 3.2.: Score distribution plots for the plain (Equation 3.3) and time penalized cosine (Equation 3.1 using Equation 3.3 as pairwise mass spectral similarity function) between all binned mass spectra from two related chromatograms. The retention time penalty of $D = 50$ s reduces the number of candidate mass spectra from a few hundred thousand to a few hundred.

illustrated in Figure 3.3, overlayed with the Gaussian retention time penalty function and a maximum retention time difference cutoff (triangles).

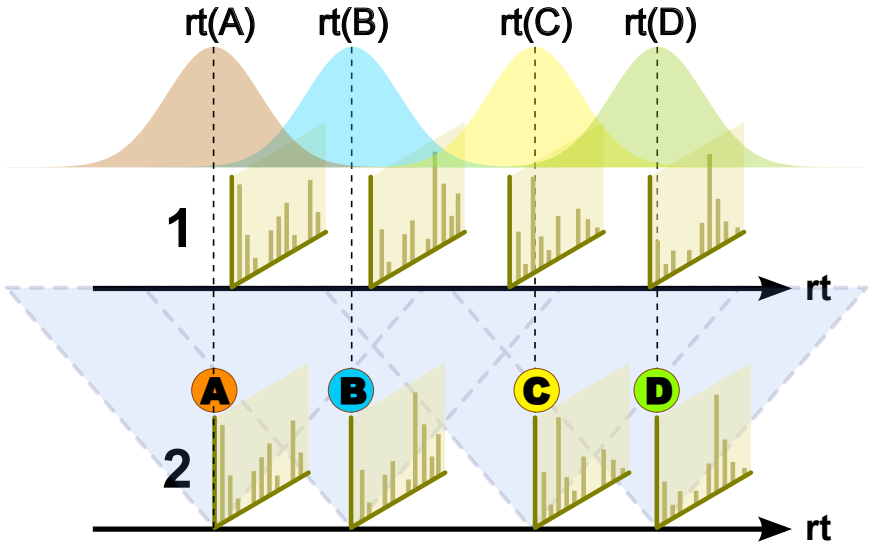
The time required to calculate all pairwise similarities between peak candidates within the different chromatograms can be reduced by using a cutoff for the maximum allowed time deviation. This is achieved by first calculating the time deviation penalty, whose value ranges between 0, indicating a large RT difference, and 1 for perfect RT correspondence, and then deciding, based on that value, whether the proximity indicates a good candidate to go on and calculate the cosine score. However, the overall complexity for this first step remains quadratic in the number of peaks to be compared.

Apparently, the simplification should only be applied if the RT deviation between two chromatograms is expected to be within a fixed time tolerance and as long as the order of elution of compounds is roughly preserved locally. Otherwise, potential candidates are pruned too early from the search space. Other similarity functions than f may also be applicable for some datasets. However, our experiments show that f gives the best overall performance on undeconvoluted spectra with low mass resolution.

We additionally employ a maximum global RT difference which should be larger than the maximum expected RT deviation. For peaks with a larger retention time difference, f will then not be evaluated at all. As stated before, this should be chosen



(a) BIPACE with a Gaussian retention time penalty function for peaks A through D from chromatogram 1 to chromatogram 2.



(b) BIPACE with a Gaussian retention time penalty function for peaks A through D from chromatogram 2 to chromatogram 1 (reverse direction).

Figure 3.3.: Schematic of the forward and reverse similarity calculation phase of BIPACE. The hard retention time difference limit is depicted by shaded cones with dashed outline. Individual Gaussian retention time penalty functions are mean centered on each peak's apex retention time (rt).

with care and generally, the global maximum retention time difference should be several times larger than D, since D is the standard deviation parameter of the Gaussian density function.

3.3.1. Assignment of Peak Pairs

We calculate the pairwise similarities using f as defined above for all possible pairs of peaks (vertices) from K different chromatograms $C_1, C_2, ..., C_K$ (partitions). Two arbitrary vertices p and q from different partitions are always distinct and never identical, but can otherwise have equal attributes, like equal masses, intensities, and retention times. Thus, the intersection of two non-identical partitions contains no common vertices. This allows us to define a K-partite edge-weighted similarity graph $S = (V, E, w)$ as follows:

Definition 8 (Best Hit Similarity Graph)**.** *Let S be a K-partite, weighted, directed graph with vertex set*

$$V = \bigcup_{r=1}^{K} C_r, \quad C_r \cap C_s = \emptyset, r \neq s \ , \tag{3.8}$$

and edge set

$$E \subseteq \{(p,q) \mid (p \in C_r,\ q \in C_s) \wedge r \neq s\} \ , \tag{3.9}$$

and the edge weight function

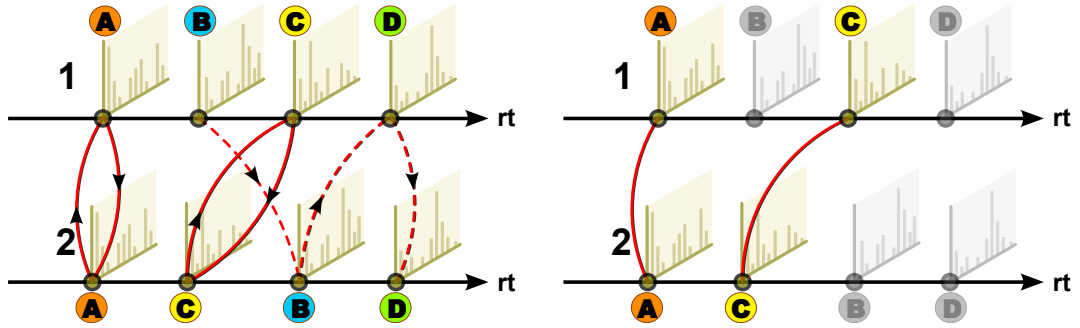
$$w(e) = f(p,q) \ , \tag{3.10}$$

for every pair of vertices (p,q) from distinct partitions. $f(p,q)$ is the pairwise similarity function that was introduced in Equation 3.1. The maximum vertex degree $d(v) \leq 2(K-1)$ for any vertex $v \in V$ such that for each pair of vertices, E contains at most one directed edge $e(p,q)$ connecting p to q, and at most one directed edge $e(q,p)$ connecting q to p.

Definition 9 (Bidirectional Best Hit)**.** *Let $p \in C_r$ and $q \in C_s$ be an arbitrary pair of vertices from distinct chromatograms C_r and C_s. Then, let q' be the vertex with highest similarity to p in C_s and let p' be the vertex with highest similarity to q in C_r.*

If $p = p'$ and $q = q'$, then p and q are bidirectional best-hit (BBH) of each other.

An example of S is illustrated for two chromatograms in Figure 3.4(a), showing directed best hit matches (weights omitted) between vertices, of which some are part of potential BBHs, namely A–A and C–C in chromatograms 1 and 2. In order to find and report all vertex groups of maximal size, spanning as many partitions as possible, we want to enumerate all maximal cliques of S. On a graph with an arbitrary vertex degree, this problem is related to the classic *NP*-complete problem *CLIQUE* (Karp 1972). There is currently no known algorithm that can solve *CLIQUE* in polynomial time for the size of the input, here the number of peaks, unless $P = NP$. We will show in Section 3.3.2 how this problem can be solved in polynomial time on a restricted graph that is derived from S.



(a) Initial hits after pairwise similarity calculation. Peaks A and C are BBHs of each other, the best hit of peak B in chromatogram 1 is peak B in chromatogram 2, but that peak has a different best hit in chromatogram 1, namely peak D.

(b) Reduced set of peak vertices and BBH edges used in the construction of graph S'. Peaks A and C are BBHs of each other and thus remain in the vertex set of S'. The edge set of S' only contains unweighted and undirected edges corresponding to the BBHs.

Figure 3.4.: Examples of graphs S and S' for two chromatograms: (a) after initial pairwise similarity calculation and (b) after reduction to the peak vertices that are part of a BBH.

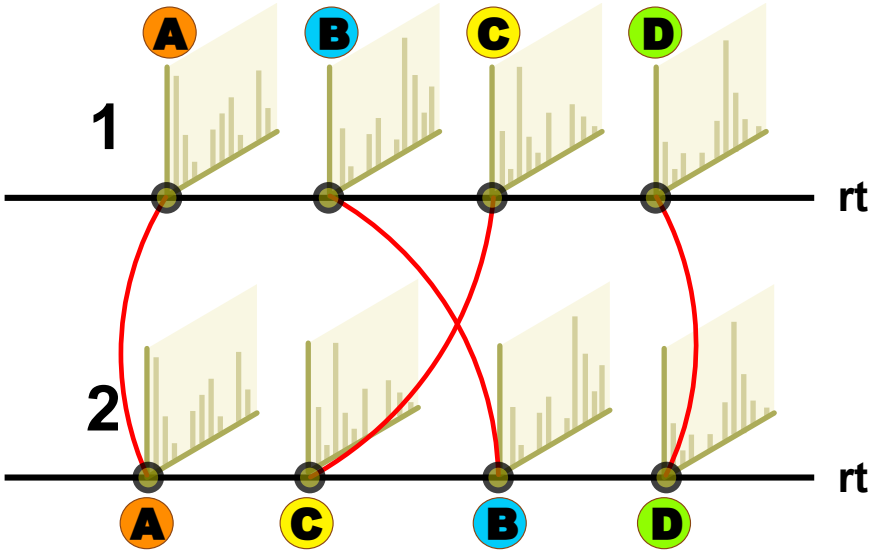


Figure 3.5.: Peak order inversion can be handled by BIPACE locally, within the defined retention time difference window.

Pruning and Performance Considerations Since only the similarities between peaks of different chromatograms are considered by our algorithm, we do not calculate the self-similarity of peaks from the same chromatogram, which differentiates our method from the method of Styczynski et al. (2007) and allows us to neglect all edges within partitions. Additionally, we exclude edges from S if they are outside the maximum retention time difference window as defined by D, which further reduces the candidate space for peak matching, but may exclude valid peak assignments. Figures 3.3(a) and 3.3(b) illustrate this for two peak lists. Nonetheless, within that window, BIPACE can handle local inversions of the elution order of peaks, as shown in Figure 3.5. This is simple to see, if only the mass spectral similarity is considered, yet it is still possible if the Gaussian retention time penalty is used as well. Here, the mass spectra of peaks B and C in chromatograms 1 and 2 occur in inverted order, but can still be assigned correctly by the algorithm based on the BBH criterion. If BIPACE used a nearest-neighbor search, it would not be possible to correctly assign peaks in such a way.

The BIPACE algorithm's pairwise similarity calculation and BBH finding phases can be implemented independently from the clique finding phase of BIPACE and can thus easily be executed individually for each pair of chromatograms, making them available for large scale parallelization.

3.3.2. BBHs Merging

In order to identify all bidirectional best hits, that are all cliques of size 2 of S, we look up for each pair of vertices $p \in C_r$ and $q \in C_s$ from distinct partitions C_r and C_s, whether they are BBHs of each other, following Definition 9. We repeat this process until all vertex pairs have been evaluated.

We then define V' as the subset of all vertices that are part of at least one BBH, and E' as the edge subset containing all BBHs, and define S' accordingly:

Definition 10 (BBH Graph)**.** *Let S' be a k-partite, unweighted, undirected graph with vertex set*

$$V' = \bigcup_{r=1}^{K} C'_r, \quad C'_r \subseteq C_r \in V(S)\ , \tag{3.11}$$

such that C' is a subset of C of S with all non-BBH vertices removed, with maximum vertex degree $d(v) \leq K - 1$ for every $v \in C'$, and edge set

$$E' \subseteq \left\{(p,q) \mid (p \in C'_r,\ q \in C'_s) \wedge r \neq s\right\}\ . \tag{3.12}$$

By construction, S' contains only vertices of degree $d(v) \geq 1$, since only vertices that are part of a BBH are included in the graph and each BBH is represented by one undirected edge in S'.

An example for S' for two chromatograms and two BBHs is given in Figure 3.4(b).

Definition 11 (Clique). *A clique in a graph $G = (V, E)$ is a fully connected subgraph $G' = (V', E')$, such that its vertex set V' is a subset of V and all vertices in V' are pairwise connected by an edge $e \in E' \subseteq E$. G' is then a* complete *(sub)graph. A clique is* maximal *if it is not contained in a larger clique.*

Definition 12 (k-clique). *We define a k-clique in a K-partite BBH graph S' as a clique that contains vertices from k distinct partitions ($k \leq K$) and at most one vertex from each of the K partitions in S'. Thus, a k-clique in S' has maximum size if $k = K$.*

We now want to enumerate all maximal cliques of S', a problem that is known to be solvable in polynomial time on graphs with a polynomial bound on the number of maximal cliques contained in the graph (Rosgen and Stewart 2007). We therefore need to determine the maximum number of maximal k-cliques in S'.

Proposition 1 (Maximum number of maximal k-cliques). *The maximum number of maximal k-cliques in S' is $\binom{K}{2}\ell$.*

Proof. We want to show that the maximum number of maximal k-cliques has an upper bound for S'. Recall that each edge in S' represents one BBH. It follows that if there are only vertices of one partition contained in S' and no edges there are also no BBHs contained in S'. Thus the maximum number of maximal k-cliques equals zero for $k = 1$.

If S' contains two partitions, the size of a maximal k-clique is 2, since any clique in S' corresponds to exactly one of the edges connecting vertices between the two partitions. The maximum number of maximal k-cliques is then equal to ℓ, which is the number of edges in S'.

For any further partition whose vertices and edges we add to the graph, the number of 2-cliques (BBH) can be at most the number of edges in the graph: $\binom{K}{2}\ell$. Each 3-clique (triangle) that we find in the graph reduces this number by two, replacing three maximal 2-cliques by one maximal 3-clique. Equivalently, for higher order cliques, the total number of maximal cliques can never grow larger than the initial number of edges in the graph. □

Proposition 2 (Maximal Clique Enumeration Problem). *If the number of maximal cliques in a graph is limited by a polynomial $p(n)$, where $n = |V|$, then the maximal cliques can be enumerated in $\mathcal{O}(nmp(n))$ time, with $m = |E|$ (Rosgen and Stewart 2007).*

Proof. For S', $p(n) = \binom{K}{2}\ell$, $m = \binom{K}{2}\ell$, and $n = K\ell$. Thus, the maximal cliques of S' can be enumerated in time proportional to

$$\mathcal{O}(nmp(n)) = \mathcal{O}(K^2\ell\ K^2\ell\ K\ell) = \mathcal{O}(K^5\ell^3)\ . \tag{3.13}$$

□

This is an upper bound for general unweighted and undirected graphs with a limited maximum vertex degree, allowing arbitrary edges between vertices, so we are able to improve it for the K-partite BBH graph defined in Definition 10. Observe

that a clique in S' is quite restricted. The vertex set of each clique $\mathfrak{C}$ must consist of vertices from disjoint partitions, meaning that for any pair of vertices $(v, w) \in \mathfrak{C}$, with $v \in C_i$ and $w \in C_j$, $i \neq j$, there exist exactly one edge in the edge set E' of S'.

Thus, the maximum search depth that we need to explore for any vertex v in S' is 1, since we only need to consider direct neighbors of v, excluding v itself (open neighborhood). There are at most $K-1$ neighbors to explore for each vertex in S', and a vertex can be a member of at most $K-1$ independent cliques, which are then of minimal size.

If we use the Bron-Kerbosch algorithm (Bron and Kerbosch 1973), essentially performing a depth-first search of cliques, starting from a given vertex, to enumerate all maximal cliques, it will run in time proportional to the number of maximal cliques contained in S'. The maximal number of maximal cliques in S' must always be smaller or equal to the number of BBHs (2-cliques) in the graph. Since for each vertex we can have at most $K-1$ BBHs, and we have $|V'| = K\ell$ as the number of vertices, the number of BBHs equals $\binom{K}{2}\ell$ and thus is smaller than $K^2\ell$. Thus, for a candidate clique C of size $K-1$ and a candidate vertex to be added to that clique, we can check in $K-1$ time, if it is compatible with all vertices already in the clique. Additionally, there must not be another peak already in the clique from the same partition, which allows us to terminate the compatibility testing early, by checking whether the target partition contains a vertex from the same partition as the candidate vertex. If not, we can continue to extend the clique.

All vertices that have not been assigned to a BBH in this phase are optionally reported as *unmatched* by the algorithm for downstream inspection.

We proceed greedily by trying to merge each pair of BBHs into a clique containing at least 2 and at most K vertices, where $K \geq 2$ is the minimal clique size (MCS) parameter. Merging is only performed if the new group of vertices remains a complete subgraph, which is equivalent to all vertices within the cluster being BBHs of each other. Otherwise, we select the largest common fully connected subgraph and omit all vertices that are not fully connected. The omitted vertices are reported as *unassigned* if they have to be removed from an otherwise complete clique and are thus singletons, and as *incompatible* if the vertices are still part of a larger clique. We continue merging until all BBHs have been processed. Finally, we report cliques with at least k vertices, ordered by the median retention time of their corresponding peaks in a multiple alignment table. Peaks whose vertices are not included in any of the final cliques are optionally reported in the AMDIS-compatible MSP format (Stein 1999) with their mass spectrum, retention time, originating file, and a unique ID for manual inspection. The clique finding is illustrated for three chromatograms and a limited number of peaks in Figure 3.6(a) for a maximal bidirectional-best hit clique and for a non-maximal clique with one not completely connected peak in Figure 3.6(b).

One requirement of the multiple alignment output is that each peak be covered at most once. Thus, if a peak is part of multiple cliques, we select the largest clique to be reported. However, this clique partitioning can be a hint that the BBH criterion is sometimes too strict and can lead to false negatives. This could be circumvented by taking not only the best hit for each peak into account, but by considering a larger

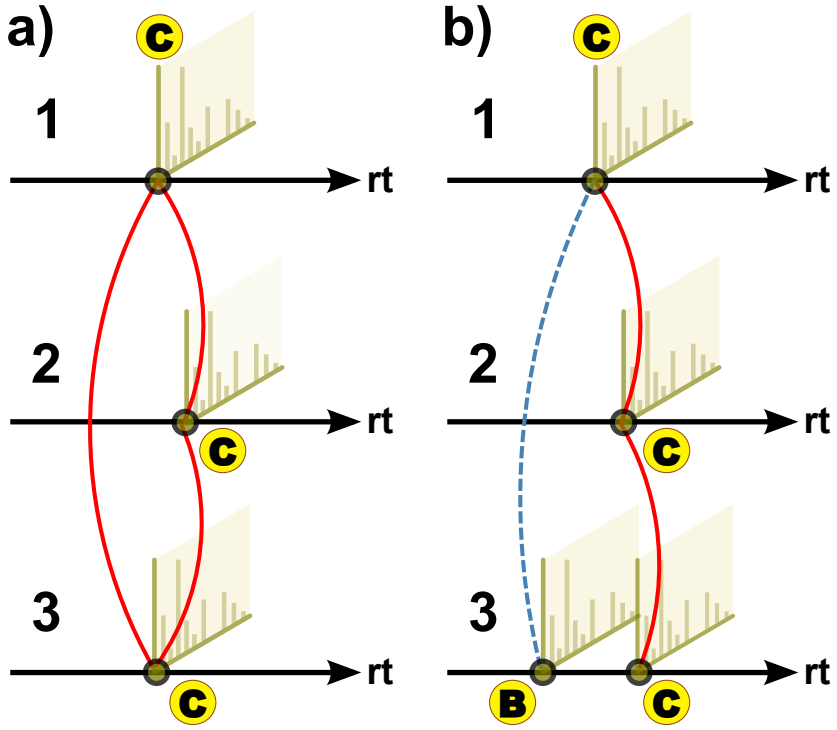


Figure 3.6.: Cliques after BBHs have been evaluated with BIPACE. (a) shows a complete clique of BBHs of peak C in all three chromatograms. (b) shows an incomplete case, where peak C in chromatograms 1 and 2, and in chromatograms 2 and 3 has a BBH. However, peak B in chromatogram 3 is only a BBH of peak C in chromatogram 1, destroying the possible complete clique of BBHs between peak C in all three chromatograms.

best hit list for each peak. Unfortunately, this would result in dramatically increased runtime and memory requirements and would render the method useless for realistic problem sizes of a few hundred chromatograms with a few thousand peaks each. An alternative is to modify the clique merging phase of the algorithm, to allow for a configurable percentage of missing BBHs. With a reasonably low value of $< 5\%$, this leads to more complete cliques, but at the same time increases the risk of finding false positives in otherwise correct cliques (data not shown).

3.3.3. Time and Space Complexity of BIPACE

We need $\binom{K}{2}\ell^2$ comparisons to calculate all pairwise peak similarities between K chromatograms with ℓ peaks each, using a symmetric similarity function $f(p,q) = f(q,p)$. Thus, the calculation of similarities requires $\mathcal{O}(K^2\ell^2)$ time and space, if we need to keep all pairwise similarities, e.g. for plotting purposes. However, we can save space by recording for every peak p from chromatogram C_i only its best hit set of size $K-1$, containing the best matching peaks $q_1, q_2, ..., q_K$, excluding q_i, where

each q_j is from a different chromatogram $C_j, j \neq i$. Then, the total size of all best hit sets is proportional to the number of peaks, $K\ell$, multiplied by the number of partitions a peak can have best hits for, $K-1$ (equivalent to the maximum vertex degree of S'), giving a total space requirement of $\mathcal{O}(K^2\ell)$ for S'.

Finding the bidirectional best hit for each peak p of the $K\ell$ peaks in S' requires that we retrieve p's best hit q and q's best hit p', and test whether $p = p'$. This amounts to $\mathcal{O}(K\ell)$ comparisons for all peaks.

In order to identify all maximal cliques, we employ a greedy, bottom-up approach based on the BBHs of each peak. Storing all BBHs clearly requires $\mathcal{O}(K\ell)$ space. Then, for each pair of peaks (p, q) from different partitions, we try to merge their corresponding cliques. This requires checking whether all peaks in the candidate cliques P and Q are fully connected, which takes $2|P||Q|$ comparisons per pair. Since $|P| + |Q| \leq |K|$, this amounts to $\mathcal{O}(K^2\ell^2)$ time.

In total, BIPACE thus requires $\mathcal{O}(K^2\ell^2)$ time and $\mathcal{O}(K^2\ell)$ space.

3.3.4. Multiple Alignment Projection

Up to now, only the grouped peaks have been aligned, so we have a peak-based multiple alignment. For a full multiple alignment of the complete datasets, all unassigned signals should also be aligned. In this situation, one could choose to implement an approach like the one proposed by Krebs and co-workers (Krebs et al. 2006), selecting a *representative* chromatogram as alignment reference and calculating a cubic spline or other higher order polynomial, to interpolate between the aligned peaks. However, such a method can only work well if the number of aligned peaks is high and there are no large areas of unknown peak assignments in the chromatograms. To circumvent these problems, we will show in the next section how to use DTW to calculate signal assignments in between paired peaks, using the same similarity function as in BIPACE . Additionally, we show how the aligned pairwise peak groups from BIPACE, or any other peak alignment method, can be used as alignment anchors for DTW, before using the pairwise DTW scores to automatically select a reasonable alignment reference using the center-star heuristic.

3.4. CEMAPP-DTW

In this section, we introduce an improved version of DTW for series of time-resolved feature vectors, as they occur in GC-MS and LC-MS data processing. In (Hoffmann and Stoye 2009), we described how to speed up DTW using predefined anchors of features which could be matched a priori with high confidence, while still allowing the alignment flexibility by defining a neighborhood radius r around the positions of the anchors. Here, we extend this approach and show how anchors can also be combined with other constraints, such as the Sakoe-Chiba Band constraint (Sakoe and Chiba 1978) to save both execution time and space, using an optimized data structure for alignment matrix storage.

Pairwise DTW is a global alignment of two series $A = (a_1, a_2, ..., a_M)$ and $B = (b_1, b_2, ..., b_N)$ of lengths M and N, respectively, where a_i,$b_j \in \mathbb{R}^L$ are the individual feature vectors of equal dimension L. In the context of GC-MS and LC-MS, a feature vector corresponds to a binned mass spectrum of intensities, a base peak ion intensity or a TIC value. We assume that mass resolution and range are equal for the experiments to align, thus only the intensity distribution over a fixed range of mass channels is used as feature vector.

The common definition of DTW involves a local distance function and a global distance or *objective* function that should be minimized (Clifford et al. 2009). To be consistent with our previous notation, we use an equivalent formulation using similarities, which then requires maximization of the objective function. Since A and B are series sampled at discrete intervals, we seek an optimal matching of elements (i, j) connecting every element in A to at least one element in B and vice versa, termed a *path* or simply *alignment*. In order to find an optimal alignment of A and B, an $(M+1) \times (N+1)$ alignment matrix $\mathcal{Q}$ is set up, in which the optimal similarity value for aligning the prefixes $(a_1, ..., a_i)$ and $(b_1, ..., b_j)$ is stored at position $\mathcal{Q}(i, j)$. A path $\mathcal{P} = (p_1, ..., p_K)$ thus consists of elements $p_k = (i, j)$, where the path length K is bounded by $1 \leq K < 2 \cdot max(M, N)$ for non-empty A and B.

Pairwise DTW usually performs a global alignment of two series of features, requiring that the start and end of both series have to be aligned: $p_1 = (1, 1)$ and $p_K = (M, N)$. However, this constraint can be relaxed for subsequence matches to gain the equivalent of a free-end gaps alignment (Prince and Marcotte 2006). Note that DTW allows mapping of an element to multiple counterparts, which differentiates it from classical sequence alignment, where an element can only map to at most one counterpart (Kruskal and Liberman 1983). Additionally, a continuity constraint requires that $\mathcal{P}$ must move only to directly adjacent cells of the alignment matrix vertically, horizontally or on the diagonal, such that if $p_k = (i, j)$, and $p_{k+1} = (i', j')$, then $i' - i \leq 1$ and $j' - j \leq 1$ must hold. A third constraint requires monotonicity of the path, such that $i' - i \geq 0$ and $j' - j \geq 0$ hold, and $(i' - i) + (j' - j) > 0$.

An optimal alignment path satisfying the above constraints maximizes the sum of pairwise similarities. This allows us to define the optimal DTW alignment between non-empty A and B through the following expression:

$$DTW(A, B) := \max_{\mathcal{P} \in \mathbb{P}(A,B)} \left(\sum_{p_i \in \mathcal{P}} \mathcal{Q}(p_i) \right), \tag{3.14}$$

where $\mathbb{P}$ is the set of all possible global alignment paths of A and B.

Maximization alone would favor the highest number of steps to align A to B, given the above constraints, resulting in alternating combinations of vertical (*expansion*) and horizontal (*compression*) steps. Hence, Kruskal and Liberman (1983) introduce additional weighting factors to treat diagonal (*match*), vertical and horizontal steps equivalently. *Expansion* and *compression* are similar to *insertion* or *deletion* in classical sequence alignment. We thus define three weight parameters, w_{match}, w_{comp} and w_{exp},

which allow to vary the degree of flexibility of the alignment between over-adaptation and the shortest possible alignment.

Finding an optimal warping path to actually recover the mapping between A and B can be achieved by applying the dynamic programming principle and tabulating intermediate optimal results. We thus calculate the value of each $\mathcal{Q}(i,j)$ by applying Equation 3.15, with f corresponding to the same similarity function as used in Section 3.3. Initialization of row 0 and column 0 with $-\infty$ is required to only allow a global alignment, effectively forcing the alignment of (a_1, b_1):

$$\mathcal{Q}(i,j) := \begin{cases} 0 & \text{if } i = j = 0, \\ -\infty & \text{if } i = 0 \text{ and } 0 < j \leq N, \\ -\infty & \text{if } j = 0 \text{ and } 0 < i \leq M, \\ \max \left\{ \begin{array}{l} \mathcal{Q}(i-1, j-1) + w_{match} f(a_i, b_j) \\ \mathcal{Q}(i, j-1) + w_{comp} f(a_i, b_j) \\ \mathcal{Q}(i-1, j) + w_{exp} f(a_i, b_j) \end{array} \right\} & \text{for } 1 \leq i \leq N, 1 \leq j \leq M. \end{cases} \tag{3.15}$$

The optimal score can then be found in the bottom-right entry of the alignment matrix $\mathcal{Q}$, such that $DTW(A,B) = \mathcal{Q}(M,N)$. This also forces a_M and b_N to be aligned. Since we introduced the weights to artificially balance the number of expansions and compressions with respect to the number of diagonal steps, we correct the calculated score by subtracting the weights for each step of the alignment path and normalize it by the length of the path to a value between 0 (no similarity) and 1 (maximum similarity). This allows to compare series of different lengths if the same similarity function and path weights have been used (Prince and Marcotte 2006).

3.4.1. Postprocessing - Obtaining Bijective Maps

As described by Prince and Marcotte (2006), the obtained map from DTW may not be bijective, depending on the similarity function used. They present a method to select bijective anchors as control points for a polynomial fit, in order to interpolate in between the anchors. In CeMAPP-DTW, however, we choose to define path weights that either boost diagonal moves by user-definable factors, resulting in a less or more adaptive alignment path. For symmetric DTW, these factors can be used to efficiently reduce the problem of over-adaptation of the path, when maximizing a similarity function and avoiding the need to predetermine additional gap penalties. CeMAPP-DTW reports a list of the maxima of the similarity function found along the alignment trace, which coincide with aligned, highly similar mass spectra.

3.4.2. An Efficient Datastructure for Pairwise DTW Alignment with Anchors

The unconstrained pairwise DTW algorithm requires $\mathcal{O}(N^2)$ time and space, where N is the number of feature vectors to be compared. Additionally, due to the pairwise similarity used, the method requires another factor of L for each pairwise similarity

calculation. For long feature vectors, L may be larger than N. However, most regions of the calculated pairwise similarities are never needed in practice, as chromatograms tend to be distorted most around the diagonal of such a pairwise similarity matrix. In practice, the Sakoe-Chiba band (Sakoe and Chiba 1978) or the Itakura parallelogram (Itakura 1975) constraints are often used to prune regions that are too far away from the diagonal.

These constraints still do not capture the chromatographic reality, where retention time distortion is mostly caused by large peaks eluting from the column, shifting all subsequent peaks by a nonlinear factor (Podwojski et al. 2009). We therefore introduced easily identifiable peaks as anchors to DTW (Hoffmann and Stoye 2009). These anchors define regions within which the alignment is calculated exactly, whereas outside of these regions no calculations are performed at all. In order to implement this idea, here we introduce a partitioned array data structure to store only those elements that are contained in the anchor-constrained regions. This requires the previous association of anchors, e.g. by BIPACE or other methods.

Efficient Storage of Partitioned Array. We use the row compressed storage (RCS) technique to store all elements of an alignment matrix in a linear array d, where each element is accessed via an offset index array idx for each row in the virtual matrix. An element of the virtual matrix at row i and column j can be accessed using the index $k = idx(i) + j$ in array d. Iteration for virtual row i can be performed from $idx(i)$ to $idx(i) + j, j < idx(i+1) - idx(i)$. Query of elements outside of the defined regions returns a configurable default value, such as positive or negative infinity. Setting of such elements has no effect, since the layout is static and determined before initialization of the matrices.

Layout Calculation. The layout of the partitioned array is determined by explicit constraints, regarding the elements that require evaluation during the alignment. These constraints are defined by geometric primitives within the 2-dimensional plane, e.g. rectangular regions defined by the alignment anchors, as well as trapezoid or arbitrary other regions. However, the layout needs to satisfy the monotonicity and continuity constraints of DTW. Thus, directly neighboring adjacent anchors and anchors with inverted order are detected and removed.

The final shape of the partitioned array is determined by the intersection of the set of constraints $\mathcal{L}$, where $\mathcal{L}$ consists of all pairs (i, j) for which the alignment is calculated. This may lead to a less optimal alignment concerning the optimization function, but allows for further speedup and smaller memory footprint. One option here is to include either a global or a local Sakoe-Chiba band constraint between successive anchors. The width w for such bands can be defined by the user either for the whole alignment matrix (global) or for every partition (local).

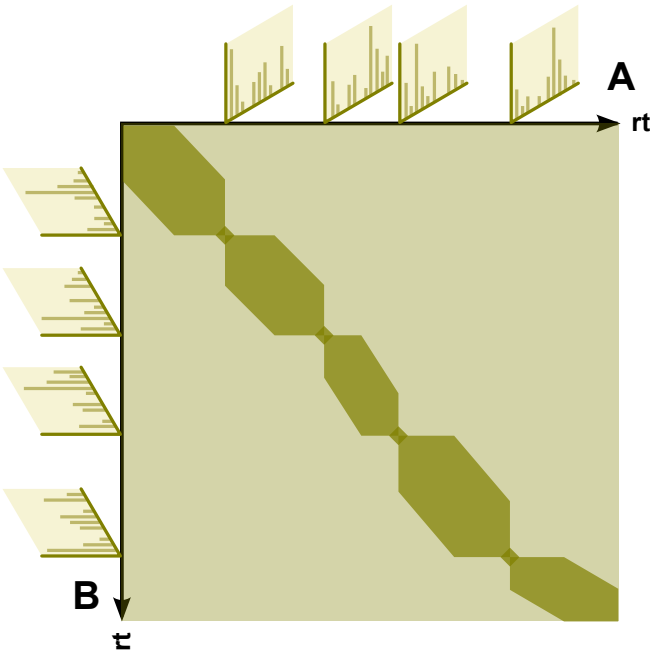


Figure 3.7.: Schematic of a pairwise alignment matrix of partitioned dynamic time warping for two arbitrary chromatograms A and B. The light shaded region represents the unconstrained alignment matrix, whereas the dark shaded areas represent the constrained partitions. For every pair of predefined anchors, in this case depicted as mass spectra, a small region around the anchor is kept to allow the alignment a higher degree of flexibility. Each partition is additionally constrained by a local Sakoe-Chiba band constraint. The intersection of all constraint sets $\mathcal{L}$ defines the final layout of the pairwise alignment matrix and thus the number of elements that are compared and stored.

We then define $\hat{\mathcal{Q}}$ as the DTW recursion to calculate $\mathcal{Q}$ using $\mathcal{L}$ as the constraint set :

$$\hat{\mathcal{Q}}(i,j) := \begin{cases} -\infty & \text{if } (i,j) \notin \mathcal{L} \\ \mathcal{Q}(i,j) & \text{otherwise.} \end{cases} \tag{3.16}$$

A schematic of the corresponding partitioned array with a constraint set $\mathcal{L}$ using anchors and a local Sakoe-Chiba band constraint is shown in Figure 3.7.

3.4.3. Multiple Alignment of Chromatograms

In order to capture machine dependent fluctuations in retention times and signal intensities, multiple chromatograms are usually measured from the same original sample as technical replicates. These often exhibit rather small, but notable, deviations in retention times and intensities, when compared pairwise.

Moreover, biological replicates show larger deviations due to the heterogeneity of the sampled population and corresponding differences in the metabolic state of cells at the time of harvesting (Christin et al. 2010).

When comparing the metabolic response of an organism under different conditions, deviations are even larger, as some metabolites may not occur at all, and others occur in different quantities, depending on the affected pathways of the organism. Thus, a multiple alignment algorithm needs to handle all of these aspects as good as possible.

Reference Selection

A general method for multiple alignment of chromatograms does not necessarily require a reference to align to. However, most published algorithms either use a manually selected reference (Chae, Reis, and Thaden 2008), or construct a reference by adding otherwise unassigned peaks (Lange et al. 2007) or by averaging over total ion chromatograms (Clifford et al. 2009). Automatic selection of a reference among the available chromatograms is seldomly reported (Christin et al. 2008) but is beneficial to methods using a manually defined reference (Robinson et al. 2007) that can introduce a bias in the process of alignment early on.

In metabolomics and proteomics applications, the number of measurements typically ranges from dozens to hundreds, such that a multiple alignment algorithm should scale well and be as memory efficient as possible, since file sizes may approach several hundred MBytes or even GBytes per raw data file. To avoid a direct multiple alignment, we calculate pairwise DTW scores between all pairs of chromatograms first. These scores can be obtained from the pairwise DTW scores, but faster methods can also be used to estimate the true scores, e.g. based on peak-matching and scoring as performed by BIPACE, although these may not be as accurate. Then, we select the chromatogram that has the highest sum of scores to all other chromatograms as the alignment reference. All remaining chromatograms are then aligned to this center chromatogram independently of each other (Hoffmann and Stoye 2009). Other authors report to use comparable clustering methods (Christin et al. 2010; Lange et al. 2007).

Multiple Alignment Construction. The construction of the multiple alignment differs slightly from the approach taken in sum-of-pairs multiple sequence alignment, since we use DTW, which is potentially a non-metric similarity function (Clote and Straubhaar 2006). Additionally, every pairwise alignment is a global alignment without gaps, so in principle we can not worsen the multiple alignment by introducing gaps. However, since DTW uses compressions and expansions, chromatograms having peaks which are absent in the selected reference may artificially decrease the quality and score of the alignment. Hence, we can not guarantee that the multiple alignment will be within a specific error bound of the optimal multiple alignment. Nonetheless, our method performs well in practice, which will be discussed in detail in the Section 3.5.

We finally obtain a dense matrix of aligned feature vector indices, e.g. of the binned mass spectra, or derived figures, such as the retention time of each mass spectrum for all chromatograms. In case of CEMAPP-DTW, and in contrast to BIPACE, there are no missing features within the table, as all features are aligned. These matrices will be used for evaluation of the alignment performance.

3.4.4. Time and Space Complexity of CEMAPP-DTW

Following the notation for time and space complexity of BIPACE, we need $\mathcal{O}(K^2\ell^2)$ comparisons to calculate all pairwise alignments between K chromatograms with ℓ mass spectra each. Using the pairwise DTW alignment similarities, we select the center chromatogram in $\mathcal{O}(K)$ time and align all remaining $K-1$ chromatograms to it in $\mathcal{O}(K\ell)$ time. If we store the pairwise alignments, they can be reused at this point, otherwise, they need to be recalculated in $\mathcal{O}(K\ell^2)$ time. Thus, the calculation of all unconstrained pairwise DTW alignments takes $\mathcal{O}(K^2\ell^2)$ in time and space.

For partitioned DTW, the runtime and space requirements for each pairwise alignment are a function of the partition length s and of ℓ. We then need $\mathcal{O}(\ell s)$ time and space to calculate each pairwise alignment. Using an additional local Sakoe-Chiba band constraint with width w, the space and time requirements for partitioned DTW are $\mathcal{O}(\ell w)$. In total CEMAPP-DTW then requires $\mathcal{O}(K^2\ell w)$ time and space.

3.5. Results

In this section, we first give a short review of existing strategies for the evaluation of peak and profile-based multiple alignment algorithms in the context of metabolomics. We then describe our approach and define useful metrics to compare alignment quality before we evaluate BIPACE and CEMAPP-DTW on two metabolomics datasets. In order to evaluate our methods we need to define what a *good* alignment is. To achieve this, we can use a ground truth of highly conserved and putatively grouped peaks, which are confirmed by MS/MS. For LC-MS in the domain of metabolomics and proteomics, such data sets were prepared and used for the evaluation of alignment algorithms (Lange et al. 2008). However, the ground truth defined by these datasets is only well defined for feature-based alignments and also requires a grouping of individual mass-to-charge ratio (m/z), RT and intensity features, which are currently not provided by either BIPACE or CEMAPP-DTW . For GC-MS metabolomics data, Robinson et al. (2007) compare their method against a ground truth defined by a human specialist.

Each alignment evaluation requires ground truth files, containing grouped features, such as triples of m/z, RT and intensity in the case of Lange et al. (2008), and simply RT in the case of Robinson et al. (2007). In the first case one scan may have multiple features, while in the second case a scan is a feature that is only identified by its RT. In order to perform the evaluation, we focused on the correctly assigned RTs and the

corresponding scan indices, since those will usually have the largest deviation across samples.

The ground truth peak group defines whether a peak is present in a sample or absent. The results of an alignment algorithm are then tested in turn against each ground truth group. If the alignment algorithm reports an aligned peak group, we count all the group's peaks that are present in the corresponding ground truth group as true positives (*TP*). Peaks that are absent in the ground truth group and in the reported peak group are counted as true negatives (*TN*). A peak that is reported as absent in the ground truth group, but as present in the alignment algorithm's reported group, is recorded as a false positive (*FP*). If a peak is reported as present in the ground truth peak group, but as absent in the reported peak group, it is reported as a false negative (*FN*). For each multiple alignment result obtained from a method, all unmatched peaks of the reference alignment, excluding absent ones, are added to the number of false negatives (FN) to normalize Recall and F1 score with respect to the size of the reference alignment.

We then use the following commonly applied measures to assess the quality of a multiple alignment:

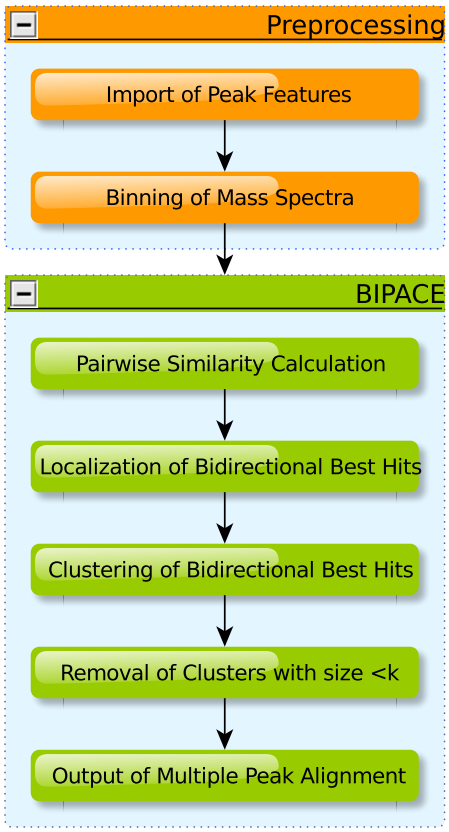
$$Precision = \frac{TP}{TP + FP}\,, \tag{3.17}$$

$$Recall = \frac{TP}{TP + FN}\,, \tag{3.18}$$

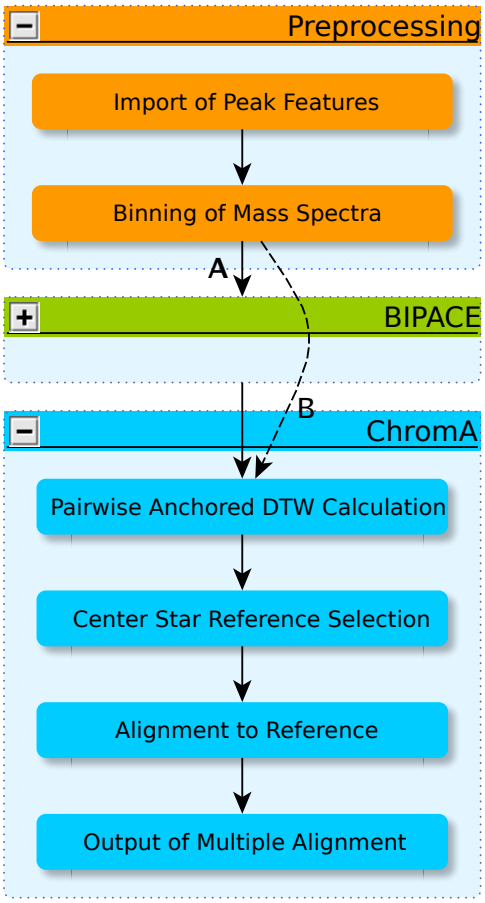
$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}\,. \tag{3.19}$$

We evaluate the performance of BiPACE and Robinson's method using precision and recall, as well as the total *TP* and *FP* numbers. For CeMAPP-DTW, however, the *TN* and *FN* values are not available, since CeMAPP-DTW reports an alignment for all peaks, so we will compare CeMAPP-DTW only using absolute *TP* and *FP* numbers.

The three major configurations that we will evaluate are schematically shown in Figure 3.8. We evaluate each of BiPACE and CeMAPP-DTW individually, before we evaluate CeMAPP-DTW using the standard BiPACE alignment with the highest *F*1 score as a constraint set. The actual alignment is preceded by a preprocessing phase, in which the peak features are imported and converted for use in our pipeline. Then, BiPACE is applied with its processing steps to calculate a multiple alignment, before CeMAPP-DTW is used first without anchors and then with the anchors as defined by the best multiple alignment of BiPACE . Throughout all evaluations, we used five different local similarities to compare the binned mass spectra, namely the cosine (*cosine*), the dot product (*dot*), the negative Euclidean distance (*euclidean*), Pearson's linear correlation (*linCorr*), and Spearman's rank correlation (*rankCorr*), each with and without a retention time penalty, as defined in Equation 3.1.



(a) Sequence of preprocessing commands for evaluation of BIPACE.



(b) Sequence of preprocessing commands for evaluation of CEMAPP-DTW. (A): with anchors; (B): without anchors.

Figure 3.8.: Workflows for the evaluation of BIPACE and CEMAPP-DTW.

3.5.1. Evaluation of BIPACE and CEMAPP-DTW on a Reference Dataset

We evaluated the BIPACE method on the *Leishmania* parasite raw data and peak lists published in Robinson et al. (2007), using as ground truth the manual multiple alignment reference from the same paper.

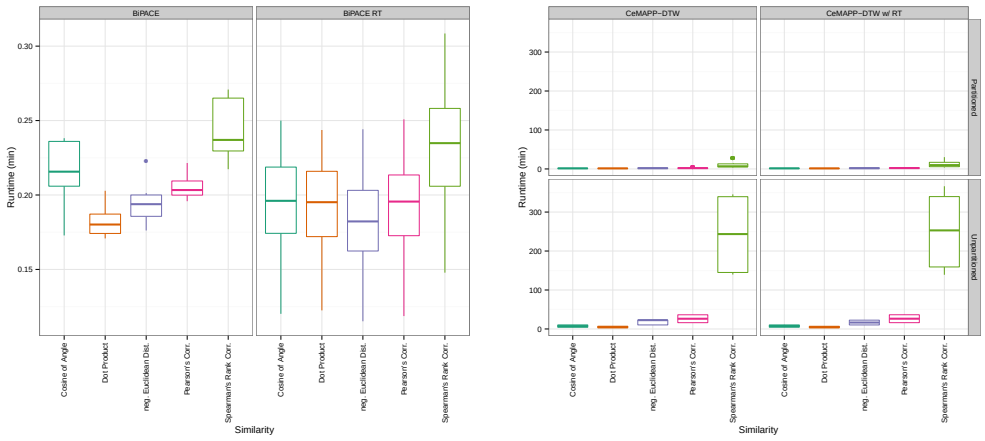
Data Preparation and Parameter Settings

Preprocessing was performed by removing intensities linked to the derivatization agent at masses 73 and 147. Due to lack of access to the manually edited peaks lists, we used the ChemStation (Agilent Technologies) peak data provided as supplementary material directly and imported them as peak annotations into our processing

pipeline. The peak data files contained between 169 and 174 peaks and were stored in tab delimited format. A line in such a file reports the apex scan index of the corresponding peak for retrieval of the raw mass spectra from the 8 different ANDI-MS/netCDF chromatogram files. Each of these files contains approximately 2780 centroided mass spectra. The spectra were binned with nominal mass accuracy in a range from 50 Da to 550 Da for further processing.

The reference manual alignment containing 173 aligned peak groups was then used in order to calculate the classification performance numbers, as defined in Equation 3.17. This was performed for each multiple alignment reported by either BIPACE or CEMAPP-DTW individually, or in conjunction, where CEMAPP-DTW used the multiple alignment of BIPACE as anchors, following Figure 3.8.

We varied the minimum clique size (*MCS*) parameter from 2 to 8 chromatograms in order to control the size of the smallest clique that should be reported by BIPACE. Other varying parameters for the time penalized instances included the width parameter D of the retention time penalty function, as defined in Equation 3.1. We also used a threshold parameter T on the value of this function so that the costly pairwise similarity function was only evaluated if the retention time penalty function's value was greater or equal to T. This pruning leads to lower runtimes of BIPACE and CEMAPP-DTW, as visualized in Figure 3.9.



(a) (b)

Figure 3.9.: Boxplots of the runtimes of (a) BIPACE and (b) CEMAPP-DTW for the *Leishmania* dataset.

For CEMAPP-DTW, we assessed two different approaches, one without any anchors from BIPACE, and one using the anchors as reported by the best BIPACE instance, as determined by the $F1$ measure. Each CEMAPP-DTW configuration was further parameterized on the weight W used for diagonal matches and on the Sakoe-Chiba band constraint BC, given as the percentage of scans from a chromatogram. For those CEMAPP-DTW instances which used the best BIPACE anchors, we

additionally varied the use of the Sakoe-Chiba band to be applied globally or locally and the size of the radius around anchors. In total, we evaluated 3106 different parametrization.

Additional figures for all parametrization together with memory usage details and a table of the best results are available in Appendix B. The raw results of this evaluation are available as supplementary file A in Hoffmann et al. (2012).

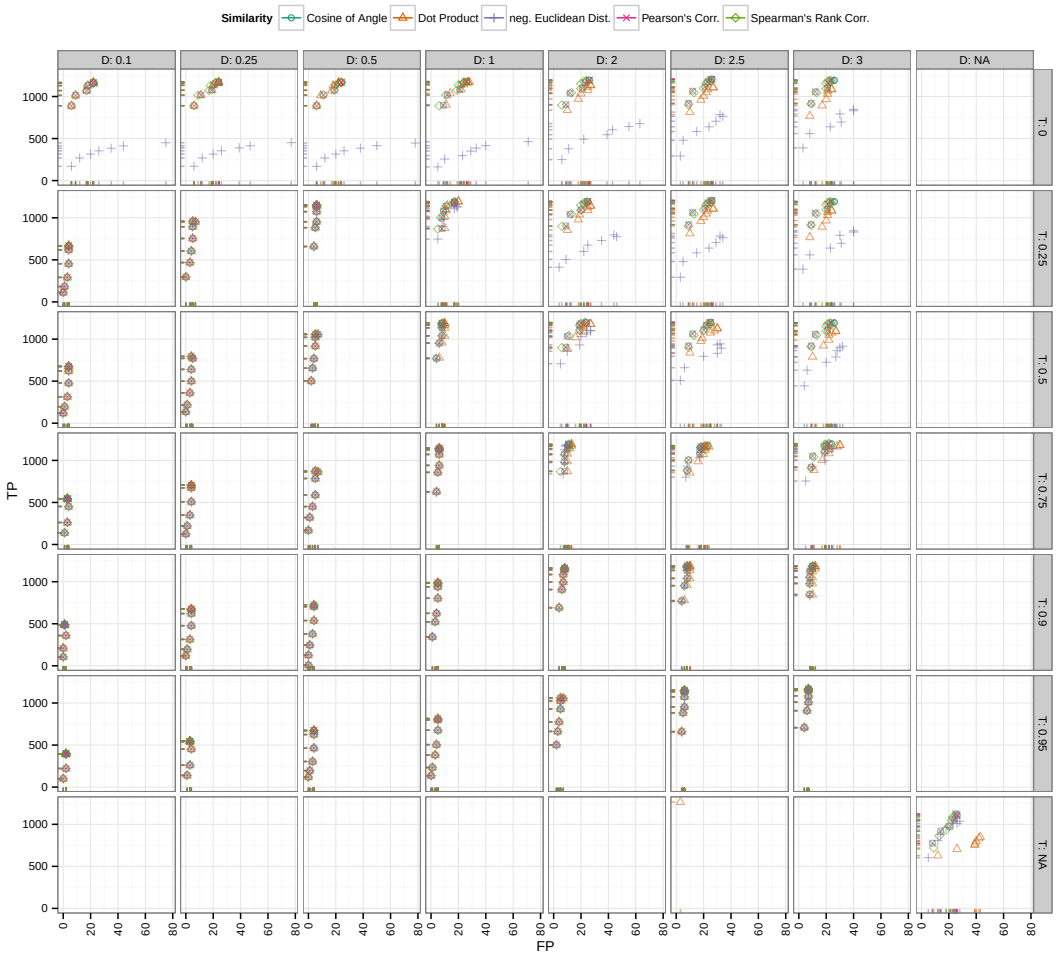


Figure 3.10.: Scatter plots for BIPACE for the *Leishmania* dataset with alignment false positives and true positives conditioned on retention time tolerance D (columns) and retention time threshold T (rows). Instances without retention time penalized similarity function are shown in the NA row/column for reference. It is visible that the unpenalized instances perform consistently worse on true positives and false positives.

Results for BIPACE

Our results for BIPACE show good performance for the time-penalized dot product, which was also used for Robinson's method (Robinson et al. 2007), but also for the time-penalized variants of Pearson's linear correlation (*linCorr*) and Spearman's rank correlation (*rankCorr*). All instances using a time-penalized variant of the similarity function are indicated in the *similarityFunction* column of Table 1 in supplementary data A of Hoffmann et al. (2012) and are shown in Figure 3.10 for varying T and D parameters. The impact of the different similarity functions on the runtime of BIPACE can be seen in Figure 3.9(a), showing that for BIPACE the runtime median was close to 38 seconds, while it was reduced for BIPACE with retention time penalty D and threshold T to less than 10 seconds. Our best result is achieved for BIPACE with Pearson's linear correlation as pairwise similarity using the time penalized variant with a minimum clique size of $MCS = 2$, T of 0.25 or 0.0 and D of 2.5 seconds. The results of the cosine similarity function are equal. For these best cases, we achieve 1206 true positives, 26 false positives, 28 false negatives and 84 true negatives. This results in a precision of 0.98, a recall value of 0.977, and a $F1$ value of 0.978. Figure 3.11 indicates that, for the best performing similarities, the choice of the MCS parameter is not critical, unless a false positive number of 0 is wanted.

Figure 3.12 shows that Robinson's result performs better than any of our parameterized instances and achieves 1264 true positives and at the same time only 3 false positives. Additionally, 3 false negatives and 114 true negatives improve the precision to 0.9976 and the recall to 0.9976, giving an $F1$ value of 0.9976. An explanation for this result can be found in our best performing alignments. There we see a larger number of false positives, meaning that our method reports more potential matches, which are scored as false positives against the given reference, but would otherwise be true positive matches. Thus, we suspect that Robinson's manually defined ground truth that we evaluate against is probably not error-free. Additionally, our best parametrizations report a number of potential aligned peak groups with significant sizes, which are not contained in the reference at all and are thus not assignable for the evaluation. If only the number of false positives is important, for example to retrieve only highly conserved peak groups with as few errors as possible, a number of parametrizations achieve that goal with 488 true positives and only 1 false positive assignment with maximum clique size of 8, a retention time threshold T of 0.9 and retention time penalty D of 0.1 s.

Results for CEMAPP-DTW

The best scoring CEMAPP-DTW result using the dot product as a pairwise similarity with diagonal match weight W of 2.25, a local Sakoe-Chiba band of $BC = 0.1$ and $D = 3$ s, using the anchors as defined by the best-scoring BIPACE instance with an anchor radius of 0 achieves 1149 true positives and 219 false positives. However, the number of false positives is potentially exaggerated since the manual reference alignment contains absent peaks, which are of course reported by CEMAPP-DTW

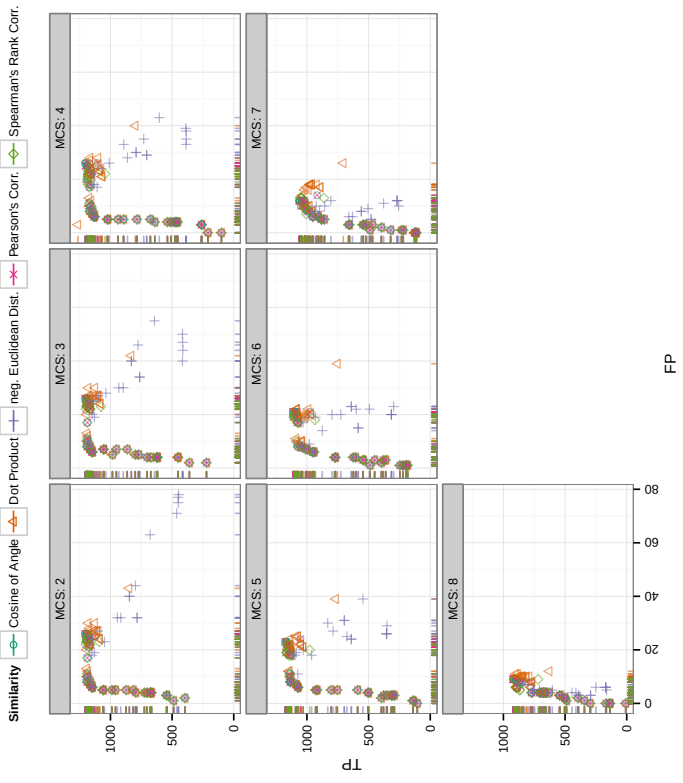


Figure 3.11.: Scatter plots for BIPACE for the *Leishmania* dataset with alignment false positives and true positives conditioned on the minimum clique size (*MCS*) parameter (columns). The highest number of true positives is found for the smallest possible value of $MCS = 2$. Fewer false positives are obtained for higher values of *MCS*, leading also to fewer true positives.

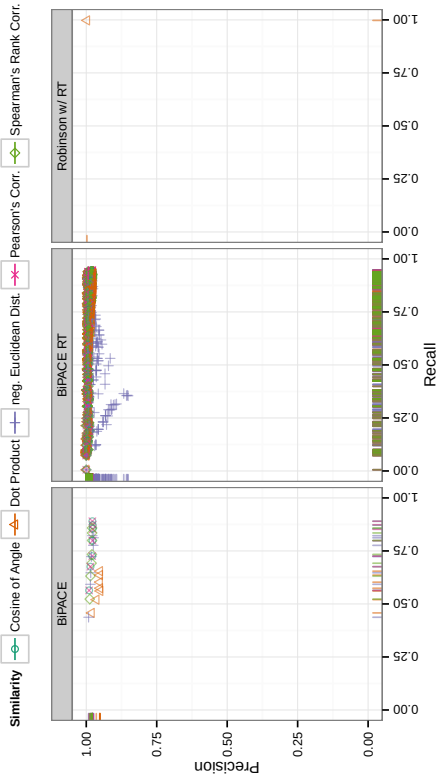


Figure 3.12.: Scatter plots for CEMAPP-DTW for the wheat dataset with alignment precision and recall. The retention time penalized variant of BIPACE performs better than the plain variant using the rank or linear correlation similarities. The published alignment of Robinson et al. (2007) performs best using a time penalized dot product similarity.

and are thus counted as false positives. The best CeMAPP-DTW result used the dot product without using anchors and a match weight of 2.25, a global Sakoe-Chiba band of $BC = 0.1$ and $D = 2.5$ and achieved 739 true positives and 549 false positives. The results for CeMAPP-DTW are visualized in Figure 3.13(a) for varying match weight W and anchor radius R and in Figure 3.13(b) for varying global or local (*BCScope*) Sakoe-Chiba band constraint BC.

3.5.2. Evaluation of BiPACE and CeMAPP-DTW on a Real World Dataset

In order to assess the quality of BiPACE and CeMAPP-DTW with and without BiPACE anchors on a GC-MS dataset of a more realistic size, we used samples from a plant metabolomics experiment (Högy et al. 2010). Spring wheat (*Triticum aestivum* L.) was grown under atmospheric and increased CO_2 concentration conditions (Högy et al. 2009) in a free-air carbon dioxide (CO_2) enrichment (FACE) field experiment. The wheat was grown, harvested, sampled at maturity in two successive years (2005, 2006), and prepared for analysis with GC-MS according to the protocol published in Högy et al. (2010) in order to determine whether the plants showed a metabolic response in their grains evident through CO_2 enrichment.

Our evaluation was based on a total of 40 chromatograms and 10 interspersed blank chromatograms. Each year was represented by 20 chromatograms, divided into two groups of 5 chromatograms each, with one technical replicate per chromatogram, summing to 10 chromatograms per condition and year. Blank runs were excluded from this evaluation. The chromatograms contained between 4615 to 4685 centroided mass spectra. The maximal scan difference that we found was around 50 scans which amounts to a maximum retention time deviation of 32 seconds between the groups of 2005 and 2006.

Data Preparation and Parameter Settings

The acquired raw data was exported using the ANDI-MS/netCDF export function of the Xcalibur software (Thermo Fisher Scientific Inc.). For all further preprocessing steps, we used our framework MaltCMS. The data was first binned along the mass axis with nominal mass accuracy by arithmetic rounding to create a dense signal matrix. Then, for each signal matrix individually, the intensities were normalized to length one for each column (binned mass spectrum) to remove linear scaling effects in intensities.

In order to assess the grouping performance, we performed a peak detection with XCMS (Smith et al. 2006), using the matched filter method with a signal-to-noise ratio of 5 and full-width at half height of 5 in order to find well represented peaks. The peak finding step reported between 410 and 465 peaks per chromatogram. The apex scan indices for each chromatogram's peaks were stored in one tab separated value file for each chromatogram.

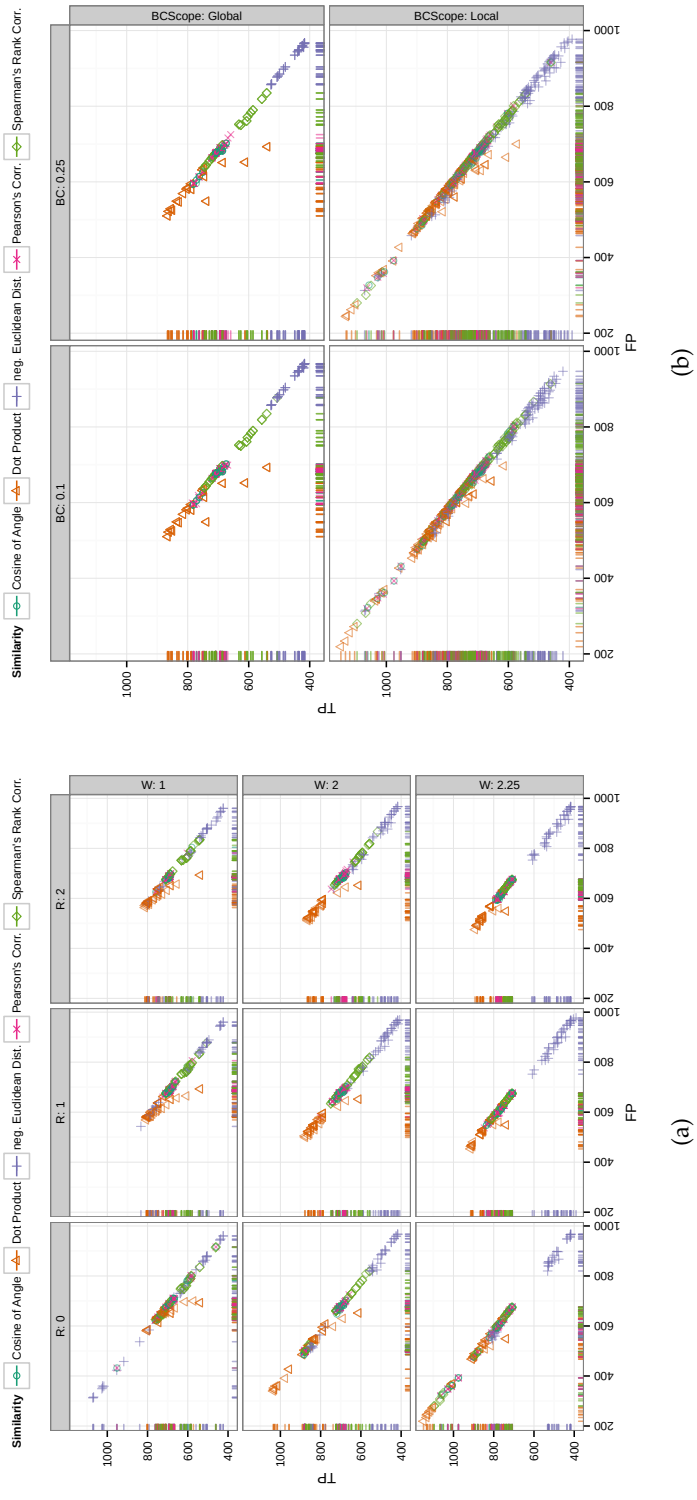


Figure 3.13.: Scatter plots for CEMAPP-DTW for the *Leishmania* dataset with alignment true positives and false positives. (a) shows alignment true positives and false positives conditioned on anchor radius (columns) and alignment match weight (rows). It is clearly visible that an anchor radius of $R = 0$ combined with a match weight of $W = 2.25$ gives the best results for linear correlation and the dot product. (b) shows alignment true positives and false positives conditioned on Sakoe-Chiba bandwidth constraint as relative number of scans (columns). Rows show whether the constraint was applied globally, indicated as *Global*, or locally, indicated as *Local*. The best results were obtained for a local window of $SC = 0.1 \cdot \max\{|A|, |B|\}$.

We then chose signals within a RT window of ±30 s. To be counted as a complete group, the scans corresponding to the tags were required to have a pairwise cosine similarity between their binned mass spectra of >0.99 throughout all chromatograms and a maximum mass deviation of 0.01 Da. The selection process lead to 184 peak groups containing peaks appearing in all chromatograms, which defined our ground truth for the evaluation of the multiple alignments produced by our methods. This reference selection and grouping was performed by a profiling method, which was recently added to MeltDB (Neuweger et al. 2008).

The evaluation was then performed following the flowchart in Figure 3.8. BiPACE was run using the raw ANDI-MS/netCDF files as input together with the tab separated value peak lists. Subsequently, the CeMAPP-DTW instances without anchors from BiPACE were run, before finally the CeMAPP-DTW instances using the BiPACE anchors from the best scoring multiple peak alignment were executed.

The reference data was then compared to the alignment results generated by the three separate evaluation workflows for BiPACE, CeMAPP-DTW, and BiPACE+CeMAPP-DTW using five different mass spectral similarity functions (dot product, cosine, linear correlation, rank correlation, negative Euclidean distance), all of them plain and in combination with a retention time penalty, as described by Robinson et al. (2007), who only report use of the time penalized dot product. We combined each similarity function with the time penalty function as in Equation 3.1.

In order to assess the precision of BiPACE, we started with a minimum clique size (*MCS*) parameter value of 40 chromatograms, meaning that only those groups that contained exactly one peak from each file were reported. For the time penalized instances we varied the width parameter D of the retention time penalty function. We also used the threshold parameter T on the value of this function so that the costly pairwise similarity function was only evaluated if the retention time penalty function's value was greater than or equal to T. The positive effect of this pruning on the runtime of BiPACE and CeMAPP-DTW is visible in Figure 3.18.

For CeMAPP-DTW, we assessed two different approaches, one without any anchors from BiPACE, and one using the anchors as reported by the best BiPACE instance, as determined by the $F1$ measure. Each CeMAPP-DTW configuration was further parameterized on the weight W used for diagonal matches and the Sakoe-Chiba band constraint width BC, given as the percentage of scans from a chromatogram. For those CeMAPP-DTW instances which used the best BiPACE anchors, we additionally varied the use of the Sakoe-Chiba band to be applied globally or locally and the size of the radius around anchors.

The exact configuration and evaluation results for all 1641 parametrizations including memory usage are available in supplementary material B of Hoffmann et al. (2012). Additional Figures and tables showing the best results are provided in Appendix B.

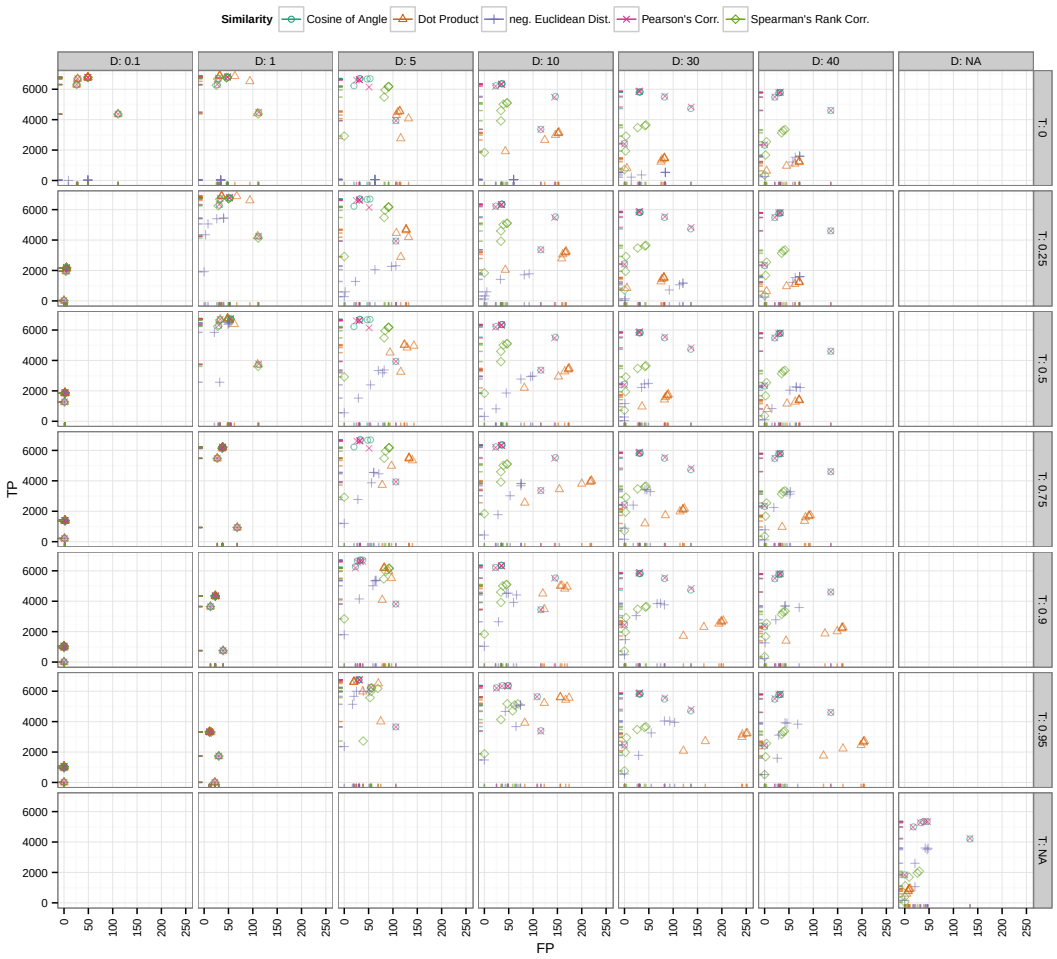


Figure 3.14.: Scatter plots for BIPACE for the *Wheat* dataset with alignment false positives and true positives conditioned on RT tolerance D (columns) and RT threshold T (rows). Instances without RT penalized similarity function are shown in the *NA* row/column for reference. It is visible that the unpenalized instances perform consistently worse on true positives, while they perform better with regard to the number of false positives.

Results of BiPACE

The results for BiPACE on the wheat dataset show very good performance in absolute and relative numbers. Figure 3.14 shows the absolute numbers of true positive versus false positive assignments for varying T (rows) and D (columns) parameters. The overall best result is achieved using the dot product (*dot*) for instances using the time penalty function, and the cosine (*cosine*) for instances not using the time penalty function. The instances using no additional retention time penalty are visible at the bottom left of Figure 3.14. These do not achieve as many true positives as the time penalized variants, however, they tend to produce fewer false positives as well. The negative Euclidean distance (*euclidean*) in combination with a time penalty, produces the fewest number of false positives, regardless of the value of D.

Figure 3.15 shows the dependency of true and false positives with regard to the *MCS* parameter. This parameter shows the relation of a small *MCS* value to a high number of true positives, but also to more false positives, since a larger number of small cliques with lower individual support are reported. Larger cliques have a high support for each contained peak and are thus more influential for the total number of true positives, but they occur less often, as is visible for $MCS = 40$, where each peak group must contain peaks from all 40 chromatograms. Again, as in Figure 3.14, dot product and cosine give the best results in absolute numbers of true and false positive assignments.

The precision and recall plot in Figure 3.16 does not clearly visualize a superior parametrization, but by inspecting the result data (supplementary material B published with Hoffmann et al. (2012)) we see, that the dot product is the best similarity function for retention time penalized instances with $MCS = 10$, 6891 true positives, 36 false positives, and 433 false negatives. The best parametrization without retention time penalty also used the cosine with $MCS = 2$, resulting in 5357 true positives, 39 false positives and 1924 false negatives. However, the retention time penalized variants tend to have a lower runtime, depending on the T parameter used.

There are no true negatives reported for the wheat evaluation, as there were no missing peak annotations in the ground truth. This explains the high number of false negatives for BiPACE, due to not completely connected peak groups, which prohibits BiPACE to form larger cliques. The peaks which could not be assigned to any cliques are consequently missing from the reported multiple alignments.

Results of CeMAPP-DTW

For CeMAPP-DTW, the results are comparable to those obtained for the *Leishmania* dataset. Without the anchors defined by BiPACE, CeMAPP-DTW has fewer true positive results and more false positive results. Here, the time penalized variant of the dot product with $D = 30$ s, BiPACE anchors, a local Sakoe-Chiba band constraint of $BC = 0.1$, and a $matchWeight = 2.25$ achieves the best result with 6459 true positives, 387 false positives and 514 false negatives.

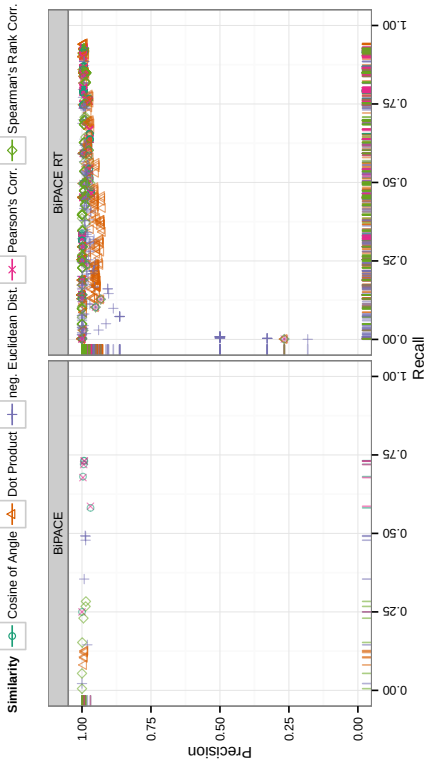


Figure 3.16.: Scatter plots for CEMAPP-DTW for the *Wheat* dataset with alignment precision and recall.

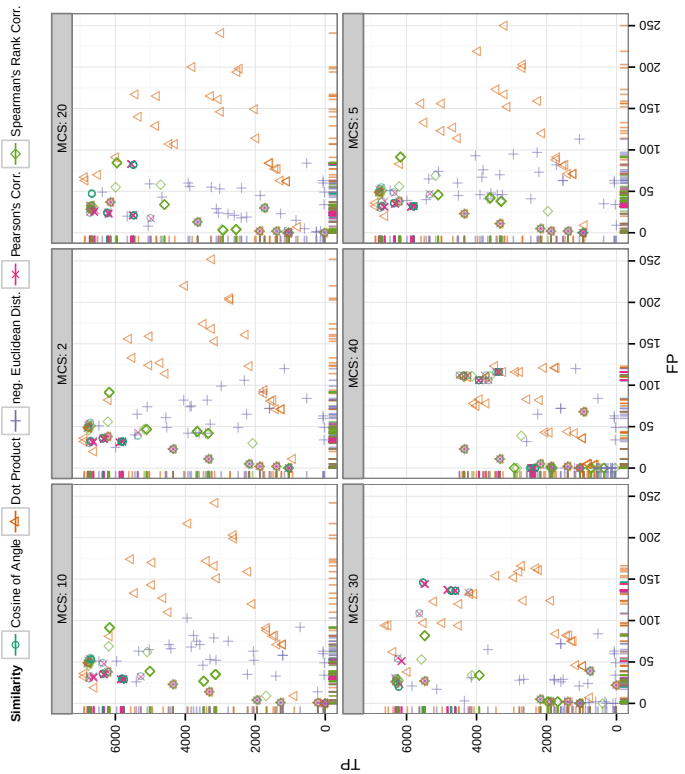


Figure 3.15.: Scatter plots for BIPACE for the *Wheat* dataset with alignment false positives and true positives conditioned on the minimum clique size (MCS) parameter (columns). The highest number of true positives is reported for the smallest possible value of $MCS = 2$. Better false positive numbers are found for higher values of MCS at the expense of true positives.

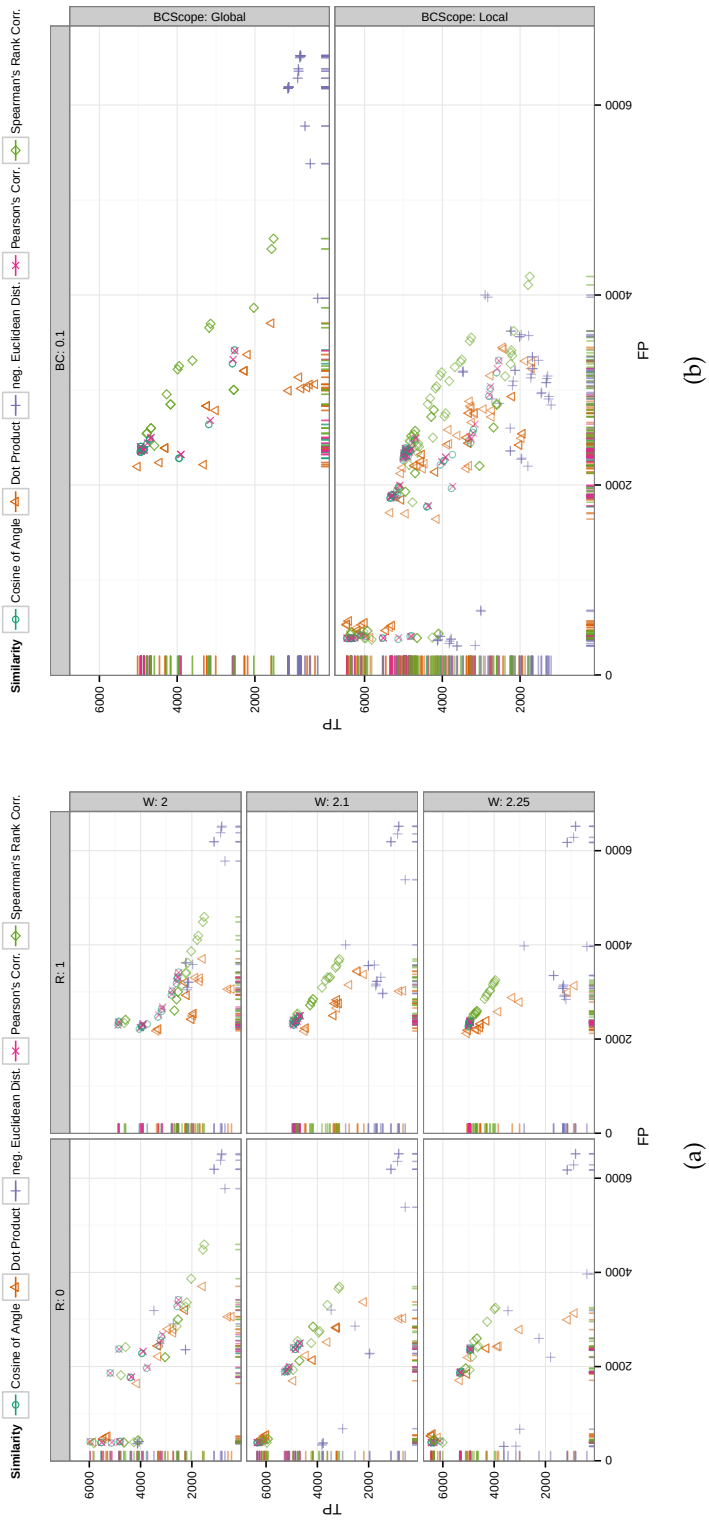


Figure 3.17.: Scatter plots for CEMAPP-DTW for the *Wheat* dataset with alignment true positives and false positives. (a) shows alignment true positives and false positives conditioned on anchor radius (columns) and alignment match weight (rows). It is clearly visible that an anchor radius of $R = 0$ combined with a match weight of $W = 2.25$ gives the best results for linear correlation and the dot product. (b) shows alignment true positives and false positives conditioned on Sakoe-Chiba bandwidth constraint BC as relative number of scans (columns). Rows show whether the constraint was applied globally or locally ($BCScope$). The best results were obtained for a local window of $0.1 \cdot \max\{|A|, |B|\}$.

The best result using no anchors from BiPACE uses the dot product with $D = 1$ s retention time penalty, a global Sakoe-Chiba band constraint of 0.1, match weight $W = 2.25$, achieving 5017 true positives, 2194 false positives and 149 false negatives. These results are illustrated in Figure 3.17, showing the dependencies of true and false positives on the different parameters. Figure 3.17(a) shows that when anchors are used to constrain CeMAPP-DTW, a small anchor radius with $R = 0$ in combination with a match weight of $W = 2.25$ provides the best results. In Figure 3.17(b) the positive effect of using a local over a global Sakoe-Chiba band width constraint with value $BC = 0.1$ is visualized and supports the claim that the local window has a positive influence on the number of true positives achieved with the anchor-constrained variant of CeMAPP-DTW.

3.6. Discussion

The results of BiPACE and CeMAPP-DTW presented in the previous sections show the advantage of using a retention time penalty as an additional criterion together with the mass spectral similarity function. The runtime boxplots in Figures 3.9(a) and 3.18(a) illustrate the advantage of using the T parameter as a threshold on the retention time penalty function. If the value of the retention time penalty function is larger than T, then the costly similarity functions are applied, otherwise, the calculation is stopped immediately for that peak pair.

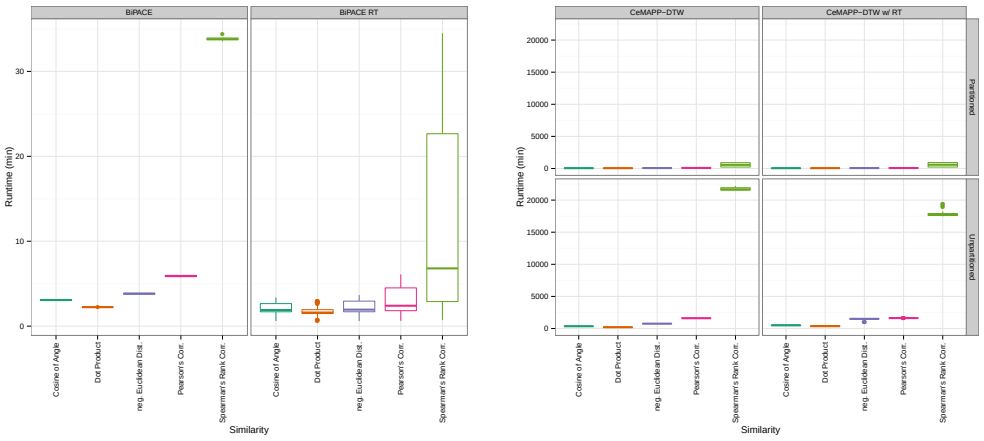


Figure 3.18.: Boxplots of the runtimes of (a) BiPACE and (b) CeMAPP-DTW for the *Wheat* dataset.

Therefore, tuning of the T parameter is one possible option to speed up the calculation of both BiPACE and CeMAPP-DTW . Since the time penalized similarity variants consistently perform better than the non-penalized ones, it is also advisable to check on the T parameter. Our results show that this parameter should initially be

set to a rather small number, since it does not directly correspond to the expected retention time deviation. Finally, the minimum clique size *MCS* is an important parameter for BiPACE and influences the number of cliques that are reported in the multiple alignment. Using a high value for *MCS* returns only those cliques whose peaks are all bidirectional best hits of each other and thus support each other as members of the clique. Lower values for *MCS* return more cliques, but at the expense of returning a higher number of smaller cliques with potentially more misaligned peaks.

CeMAPP-DTW on the other hand has a few other parameters to tune. Our results show that the most important ones are the use of anchors and an anchor radius of 0, meaning that the DTW alignment must pass through the anchor positions for example defined by BiPACE . Additionally, the use of a local Sakoe-Chiba band constraint and a match weight $W = 2.25$ are beneficial for the number of true positives CeMAPP-DTW is able to achieve.

Concerning the best similarity function to use, there is no decisive answer possible from our results. In accordance with Prince and Marcotte (2006), Pearson's linear correlation and Spearman's rank correlation give good results in terms of low false positive numbers, but time penalized dot product and cosine tend to give significantly higher true positive numbers. Using the time penalty function as a pre-filter for the actual similarity function seems to reduce the differences of the individual similarity functions. However, the instances using a correlation-based similarity have a significantly longer runtime (Figures 3.9 and 3.18) than the ones using the dot product or cosine similarity.

3.7. Conclusions

We have introduced a fast and accurate method for multiple peak alignment of GC-MS data, BiPACE, that is capable of finding groups of peaks between chromatograms that have a high similarity, achieving a high number of true positive and a very low number of false positive assignments. Our method achieves results comparable to that of Robinson et al. (2007), while being easily tunable to a very low false positive rate via the minimum clique size parameter. With the use of the peak groups aligned by BiPACE as anchors within partitioned DTW, we address one major issue of similar profile-alignment algorithms, namely their quadratic time and space complexity by partitioning the pairwise alignment matrix into adjacent regions. Thus, strong peak candidates, such as reference compounds with unique mass traces (GC-MS) or characteristic fragmentation patterns (GC-MS) are definitely aligned, while weaker peaks that were not discovered during peak finding are also aligned, but with more flexibility.

We have shown that the partitioned DTW algorithm used in CeMAPP-DTW on its own is able to calculate a profile-based multiple alignment in less time and with fewer space requirements when compared to unconstrained DTW . Combining CeMAPP-DTW with the aligned peak groups from BiPACE as alignment anchors

allowed us to improve both on the runtime, as well as on the number of true positives recovered by the alignment. This combination of the two algorithms is feasible if a definite alignment is not the main requirement, but instead the output of CEMAPP-DTW is used for a subsequent retention time correction of the profile data. For a definite multiple peak alignment BIPACE is the better alternative.

4

Methods for GC×GC-MS Data Analysis

In this chapter, we give an overview and feature comparison of existing Open Source frameworks for the handling and processing of data from comprehensive two-dimensional gas chromatography-mass spectrometry (GC×GC-MS) experiments. As in the previous chapter, this overview covers the parts of the typical processing pipeline for metabolomics data that we introduced in Chapter 2, but this time for GC×GC-MS data.

We then describe the peak finding problem for GC×GC-MS data and present a novel method to detect and filter peak seeds that can serve as input for our multiple peak alignment algorithm BiPACE 2D. We describe BiPACE 2D in Section 4.3 and evaluate it against a collection of other state-of-the-art algorithms, before discussing the results in Section 4.4. BiPACE 2D was originally published in Hoffmann et al. (2014).

The algorithms presented in this chapter are also available within our OpenSource framework Maltcms[1].

4.1. Frameworks for GC×GC-MS Analysis

The automatic and routine analysis of comprehensive GC×GC-MS data is yet to be established. GC×GC-MS couples a second chromatographic column to the first one, thereby achieving a much higher peak capacity and thus a better separation of closely co-eluting analytes (Castillo et al. 2011). Usually, for a one-hour run, the raw data file size exceeds a few Gigabytes. Quite a number of algorithms have been published on alignment of peaks in such four-dimensional (first column retention time, second column retention time, mass, and intensity values) data (Kim, Fang, et al. 2011; S. Wang et al. 2010; Vial et al. 2009; Oh et al. 2008; Pierce et al. 2005), however only a few methods are available for a more complete typical preprocessing workflow. We herein focus on frameworks that are Open Source software and that also do not

1. http://maltcms.sourceforge.net

require a proprietary software for execution, such as MATLAB (The Mathworks, Natick, MA, USA). A compact overview of the available frameworks, their licenses and programming languages is given in Table 4.1. A more detailed feature matrix of these frameworks is given in Table 4.2. The remainder of this section gives a concise overview of the frameworks Guineu (Castillo et al. 2011) and **Chrom**atogram **A**lignment for **4D** GC×GC-MS data (ChromA4D)[2].

4.1.1. Guineu

Guineu is a graphical user interface and application for the comparative analysis of GC×GC-MS data. It currently reads LECO ChromaTOF (LECO Corp., St. Joseph, MI, USA) software's peak list output after smoothing, baseline correction, peak finding, deconvolution, database search and retention index (RI) calculation have been performed within ChromaTOF.

The peak lists are aligned pairwise using the Score Alignment algorithm, which requires user-defined retention time windows for both separation dimensions. Additionally, the one-dimensional RI of each peak is used within the score calculation. Finally, a threshold for mass spectral similarity is needed in order to create putative peak groups. Additional peak lists are added incrementally to an already aligned path, based on the individual peaks' score against those peaks that are already contained within the path.

Guineu provides different filters to remove peaks by name, group occurrence count, or other features from the ChromaTOF peak table. In order to identify compound classes, the GMD substructure search is used (Hummel et al. 2010). Peak areas can be extracted from ChromaTOF using the TIC, or using extracted, informative or unique masses. Peak area normalization is available relative to multiple user-defined standard compounds.

After peak list processing, Guineu produces an output table containing information for all aligned peaks, containing information on the original analyte annotation as given by ChromaTOF, peak areas, average retention times in both dimensions together with the average RI and further chemical information on the functional group and substructure prediction as given by the GMD. It is also possible to link the peak data to KEGG (Kanehisa et al. 2013) and PubChem (Y. Wang et al. 2009) via the chemical abstracts services (CAS) annotation, if it is available for the reported analyte.

For statistical analysis of the peak data, Guineu provides fold change and t-tests, PCA, ANOVA and other methods. Guineu's statistical analysis methods provide different plots of the data sets, e.g. for showing the principal components of variation within the data sets after analysis with PCA.

2. ChromA4D is a pipeline within the Maltcms framework, please see chapter 5 and Section A.2 for details.

Table 4.1.: Open Source software frameworks for GC×GC-MS based metabolomics. a: Eclipse Public License version 1.0.

Name	Version	Methods	License	Programming language
GUINEU	0.8.2	GC×GC-MS (LC-MS)	GPL v2	Java 6
MALTCMS/CHROMA4D	1.3	GC×GC-MS (LC×LC-MS)	L-GPL v3, EPL v1[a]	Java 7

Table 4.2.: Feature comparison of Open Source software frameworks for preprocessing of GC×GC-MS based metabolomics data. Key to abbreviations: **Data formats** A: NetCDF, G: ChromaTOF peak lists, H: CSV peak lists. **Signal preprocessing** MA: moving average, MM: moving median, TH: top-hat filter, CV: coefficient of variation threshold. **Peak detection** MAX/CWT-SRG: TIC local maxima or continuous wavelet transform, followed by seeded region growing based on ms similarity. **Multiple peak alignment** SCORE: parallel iterative score-based, CLIQUE: progressive clique-based.**Visualization** (of unaligned and aligned data) TIC: plots of total ion chromatogram/peaks, EIC: plots of extracted ion chromatograms/peaks, SURF: surface plots of profile matrix (rt x m/z x I), STATS: visualization of statistical values. **DB search** GMD: Golm metabolite database webservice, PUBCHEM: pubchem database webservice, KEGG: kegg metabolite database, MSP: msp-format, compatible with AMDIS and GMD format. **Normalization** RP: reference peak area, EV: external value, e.g. dry weight. **Statistical evaluation** CV: coefficient of variation, FLT: fold-test, TT: groupwise t-test, PCA: principal components analysis, CDA: curvilinear distance analysis, SP: Sammon's projection, ANOVA: analysis of variance, FT: F-test, between group vs. within group variance.

Feature (GC×GC-MS pipeline)	GUINEU	CHROMA4D
Data formats	G	A,H
Signal preprocessing	no	MA, MM, TH, CV
Peak detection	no	MAX/CWT-SRG
Multiple peak alignment	SCORE	CLIQUE
Visualization	STATS	STATS, TIC, EIC, TIC2D
DB search	GMD, PUBCHEM, KEGG	MSP (GMD)
Normalization	RP	RP, EV
Statistical evaluation	CV, FLT, TT, PCA, CDA, SP, ANOVA	FT

4.1.2. ChromA4D

For the comparison of GC×GC-MS data, ChromA4D accepts NetCDF files as input. Additionally, the user needs to provide the total runtime on the second orthogonal column (modulation time) to calculate the second retention dimension information from the raw data files. For tentative metabolite identification, the location of a database can be given by the user. ChromA4D reports the located peaks, their respective integrated TIC areas, their best matching corresponding peaks in other chromatograms, as well as a tentative identification for each peak. Furthermore, all peaks are exported together with their mass spectra to MSP format, which allows for downstream processing and re-analysis with AMDIS and other tools. The exported MSP files may be used to define a custom database of reference spectra for subsequent analyses.

Peak areas are found by a modified seeded region growing algorithm. All local maxima of the TIC representation that exceed a threshold are selected as initial seeds. Then, the peak area is determined by using the distance of the seed mass spectrum to all neighbor mass spectra as a measure of the peak's coherence. The area is extended until the distance exceeds a given threshold. No information about the expected peak shape is needed. The peak integration is based on the sum of TICs of the peak area. An identification of the area's average or apex mass spectrum or the seed mass spectrum is again possible using the MetaboliteDB module *maltcms-db*.

To represent the similarities and differences between different chromatograms, again bidirectional best hits are used to find co-occurring peaks. These are located by using a distance that exponentially penalizes differences in the first and second retention times of the peaks to be compared. To avoid a full computation of all pairs of peaks, only those peaks within a defined window of retention times based on the standard deviation of the exponential time penalty function are evaluated.

ChromA4D's visualizations represent aligned chromatograms as color overlay images, similar to those used in differential proteomics. This allows a direct visual comparison of signals present in one sample, but not present in another sample.

ChromA4D creates peak report tables in CSV format, which include peak apex positions in both chromatographic dimensions, area under curve, peak intensity and possibly tentative database identifications. Additionally, information about the matched and aligned peak groups is saved in CSV format.

4.2. Peak Finding

We already identified different steps of a typical metabolomics processing pipeline in Section 2.6 and have discussed the implementations available in Open Source software for those steps in the previous section. One important step in such a pipeline is the location and integration of significant areas in the chromatographic domain that relate to analytes of interest, either within the TIC, or within individual EICs. This step is usually subsumed as *peak finding* and *peak integration*.

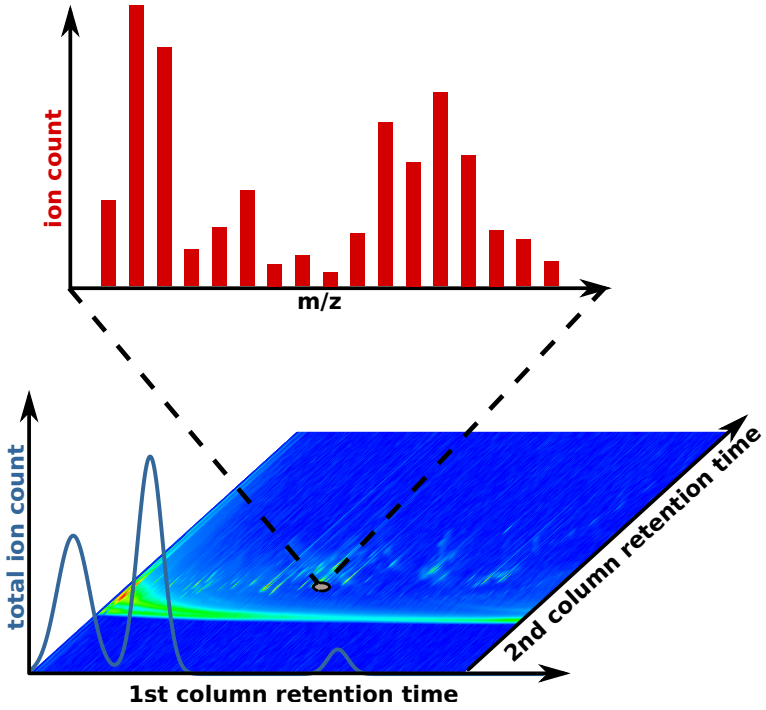


Figure 4.1.: The two-dimensional chromatographic plane in GC×GC-MS. Each point on the plane corresponds to an individual mass spectrum with first and second column retention time.

In GC×GC-MS, the chromatographic domain is a two-dimensional plane spanned by the retention times of the first and second chromatographic column, as shown in Figure 4.1. Each point on the plane corresponds to an individual mass spectrum. In principle, there are two possible approaches to peak finding: top-down and bottom-up. While the first approach starts from the (two-dimensional) TIC, identifies regions of interest and then inspects individual extracted ion traces, the bottom-up method starts at the extracted ion traces and tries to identify peaks directly within that domain. However, for high scan acquisition rates in the range of >100 Hz, this quickly leads to the inspection of many regions of the chromatogram that contain redundant information (like the peaks within the modulation peak areas). Thus, we chose the top-down approach, identifying peak candidates in the TIC and then using the individual mass spectra to determine the extents of those candidate peaks using a suitable similarity.

Preprocessing Many algorithms for peak finding and integration have been published, usually requiring some sort of previous filtering of the signal to ensure good analytical properties of the signal (Biller and Biemann 1974). Usually, the signal is smoothed using a higher-order polynomial interpolation scheme, such as the method described by Savitzky and Golay (1964) and Peters et al. (2007), LOESS-Interpolation

(Smith et al. 2006), or a signal reconstruction based on a prior decomposition into frequency components (Fourier/Inverse Fourier Transform), where high frequency and low frequency components are omitted to remove noise (high frequency) and baseline (low frequency) trends in the signal. An alternative to frequency decomposition is the scale/translation decomposition used in the continuous, maximum-overlap, or discrete wavelet transforms (Z. Zhang et al. 2012). The CWT has some desirable properties that we will address in the following sections.

Peak Location The peak location is usually determined in one of three ways: by an extremal value analysis of the signal, usually involving first, second, and third order derivatives, in order to identify local maxima, minima, and saddle points. This requires that the signal is differentiable and also explains the requirement for prior smoothing. The located maxima, minima and saddle points can then be used to fit a theoretical peak function (usually following Gaussian or Poisson distribution density functions) at the suspected location. This method is used by the peak finder in OpenChrom (Wenig and Odermatt 2010). The second method is to use a template peak (second order derivative of the Gaussian) with a specified width, convoluting it with the signal and recording the positions of maximum response, omitting all signals below a user-defined minimum threshold. This method is termed *matched filtration* and is used by XCMS (Smith et al. 2006) and related methods (Fredriksson et al. 2009). The third method uses the CWT, which does not require pre-processing of the signal. It convolutes the signal with a scaled and translated template peak function at different scales, representing suspected peak widths, and throughout the length of the signal. This method has been introduced by Du et al. (2006) for peak detection in non-centroided SELDI-TOF mass spectra. For further information, Matos, Duarte, and Duarte (2012) give a comprehensive overview of peak detection methods currently used in GC×GC.

4.2.1. Notation

In the following sections, we will define and explain the CWT, its background in signal processing, and its application to GC×GC-MS data. We define the two-dimensional chromatogram $\mathcal{C}_{2d} = \{c_1, c_2, ..., c_\ell\}$ as an ordered set of features vectors $c = (\mathbf{m}, \mathbf{i}, t_1, t_2)$, where the mass vector $\mathbf{m}$ and the intensity vector $\mathbf{i}$ both have the same dimensions, and t_1 and t_2 are the retention times of the feature vector on the first and second chromatographic column. The feature vectors are sorted in ascending order, first by t_1, then by t_2. The total ion current (TIC) of $\mathcal{C}_{2d}$ is the ordered set of the sum over each feature vector's intensity vector:

$$TIC = \left\{ x \mid x = \sum_{j=1}^{\ell} c_j(\mathbf{i}), c_j \in \mathcal{C}_{2d} \right\}, \tag{4.1}$$

where $c(\mathbf{i})$ is shorthand notation for the intensity vector of feature vector c and ℓ is the length of *TIC*. Note that the original order of the feature vectors is still preserved.

In order to bring our notation into accordance to the standard notation used in wavelet theory (Percival and Walden 2000), we define $x(t)$ as the TIC value of the t'th feature vector in TIC: $x(t) = TIC(t) = \sum_{t=1}^{\ell} c_t(\mathbf{i})$.

4.2.2. Continuous Wavelet Transform-based Peak Finding

Our goal is to locate prospective chromatographic peaks in $x(t)$, while at the same time we want to exclude areas of the signal that result from slowly changing baseline and short term spiking noise. A short one-dimensional section of a GC×GC-MS chromatogram showing peaks with varying baseline is shown in Figure 4.2. We omit the information on the second retention time dimension in the data for now, since the process of modulation (see Section 2.2.3) may introduce shifting artifacts in the second dimension of the two-dimensional TIC. We will consider the second column elution time again when we seek to filter potential peaks based on their two-dimensional neighborhoods in Section 4.2.3 and when we merge neighboring peaks in Section 5.2.3. The result of the peak finding method are peaks following our definition from Section 2.5.

We already introduced the principle of matched filtration in Section 4.2 and mentioned the continuous wavelet transform that can be seen as a generalization of repeated matched filtration for arbitrary sizes of the matching filter.

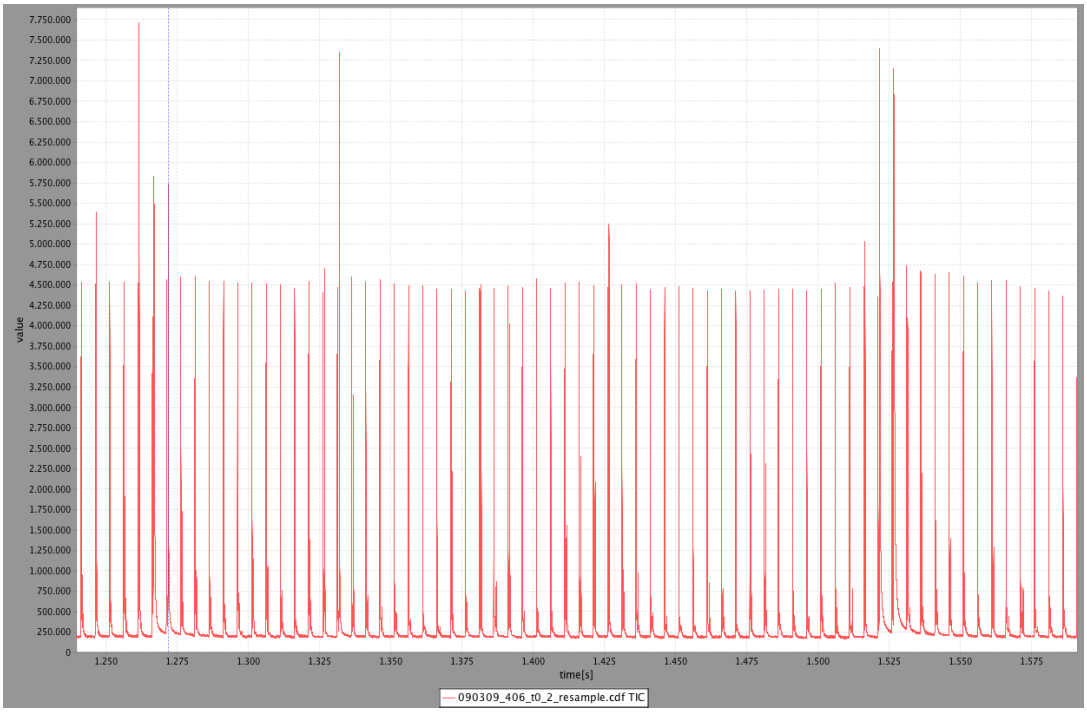


Figure 4.2.: One-dimensional section of a two-dimensional GC×GC-MS TIC.

The principle of the CWT is the projection of a given signal $x(t)$ to a space spanned by constrained normalized basis functions, the *wavelets*. The CWT requires the choice of a such a basis function in form of the *mother wavelet* $\psi(t)$. The CWT is a multiscale transform of the original signal into scale and translation dependent components. Thus, for each scale, a scaled (s, $s > 0$), translated (τ, $\tau \in \mathbb{R}$), and normalized variant of the mother wavelet, namely the *daughter wavelet* $\psi_{\tau,s}(t)$, is derived:

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right), \tag{4.2}$$

and folded with the signal:

$$CWT_x^{\psi}(\tau,s) = \frac{1}{\sqrt{s}}\int_{-\infty}^{+\infty} \underbrace{x(t)}_{signal} \underbrace{\psi^*_{\tau,s}\left(\frac{t-\tau}{s}\right)}_{daughter\ wavelet} dt\ . \tag{4.3}$$

The CWT calculates the inner product of the signal and the complex conjugated daughter wavelet $\psi^*_{\tau,s}(\cdot)$ as a basis function. The result (response) of signal and daughter wavelet at a given scale and translation is maximal if both are identical or 0 if they are orthogonal.

The mother wavelet function $\psi(\cdot)$ needs to satisfy additional properties to be amenable as a basis function suitable for the wavelet transform. It must therefore have a vanishing integral, when integrated from $(-\infty, \infty)$:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0\ , \tag{4.4}$$

and its square must integrate to 1:

$$\int_{-\infty}^{\infty} \psi(t)^2 dt = 1\ . \tag{4.5}$$

Furthermore, a wavelet is confined to a closed interval $[-T, T]$, its finite support. Within the support region its response value is maximal, but still has to cancel out (see Equation 4.4). Outside of the support region its response value is negligible, so that the majority of the wavelet's area is localized within $[-T, T]$ and an arbitrarily small ϵ may be localized outside of that interval.

This property sets it in contrast to the complex exponential functions: $e^{ix} = \cos(x) + i\sin(x)$, that are used as basis functions in the Fourier family of *change of basis* transforms. These are defined within the open interval $(-\infty, \infty)$ and are thus not localizable in their response. The complex exponential parameterized by the frequency of their basis functions and the response of folding the signal with a succession of basis functions with different frequencies produces the (complex) frequency spectrum. However, in contrast to the (continuous) wavelet transform, the Fourier transform does not allow for localization of the maximum response in the original signal from the frequency spectrum (Percival and Walden 2000).

Signal Reconstruction Both the Fourier and the wavelet transform allow a reconstruction of the original signal from the respective results of the transform in theory, however, in practice, the inverse Fourier transform requires more information (frequencies) to reconstruct transient signals to a given accuracy (Walker 1997). For the inverse CWT to allow perfect reconstruction of the original signal, the mother wavelet $\psi(\cdot)$ has to additionally adhere to the *admissibility* condition, given its Fourier transform at frequency f:

$$FT(f) = \int_{-\infty}^{\infty} \psi(t) e^{-i2\pi f t} dt, \tag{4.6}$$

so that

$$C_\psi = \int_0^{\infty} \frac{|FT(f)|^2}{f} df, \quad 0 < C_\psi < \infty\,. \tag{4.7}$$

Thus, the Fourier transformed wavelet at frequency f needs to have a non-vanishing, limited energy spectral density $FT(f)$.

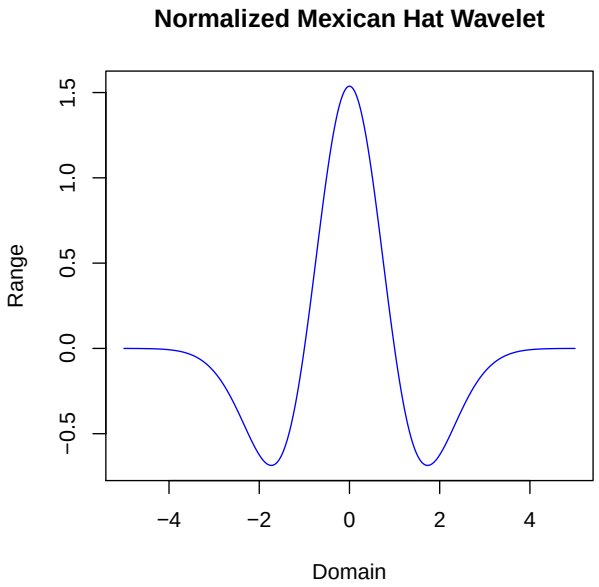


Figure 4.3.: Normalized Mexican Hat Wavelet, $\sigma = 1$.

The Mexican Hat Wavelet We have now established the necessary background for the CWT, however, we still need to choose a mother wavelet function that is applicable to short term transient data, like our TIC.

Du, Kibbe, and Lin (2006) describe a real valued mother wavelet that they use to detect peaks in single, but complex SELDI-TOF mass spectra of peptides. They use the second derivative of the Gaussian normal probability density function which is more commonly known as the Mexican Hat Wavelet (MHW):

$$\psi(t) = \frac{2}{\sqrt{3\sigma}\pi^{\frac{1}{4}}}\left(1 - \frac{t^2}{\sigma^2}\right)e^{\frac{-t^2}{2\sigma^2}} \; . \tag{4.8}$$

The shape of this mother wavelet is similar to the optimal analytical peak shape (see Figure 4.3), with the additional ability to automatically remove slowly changing background within its support region. In real data, peaks usually show tailing behavior and are thus not symmetric, but the MHW will still show a large response value even when folded with such non-perfectly symmetric peaks. By tracking the response over multiple scales, such peaks can then also be resolved.

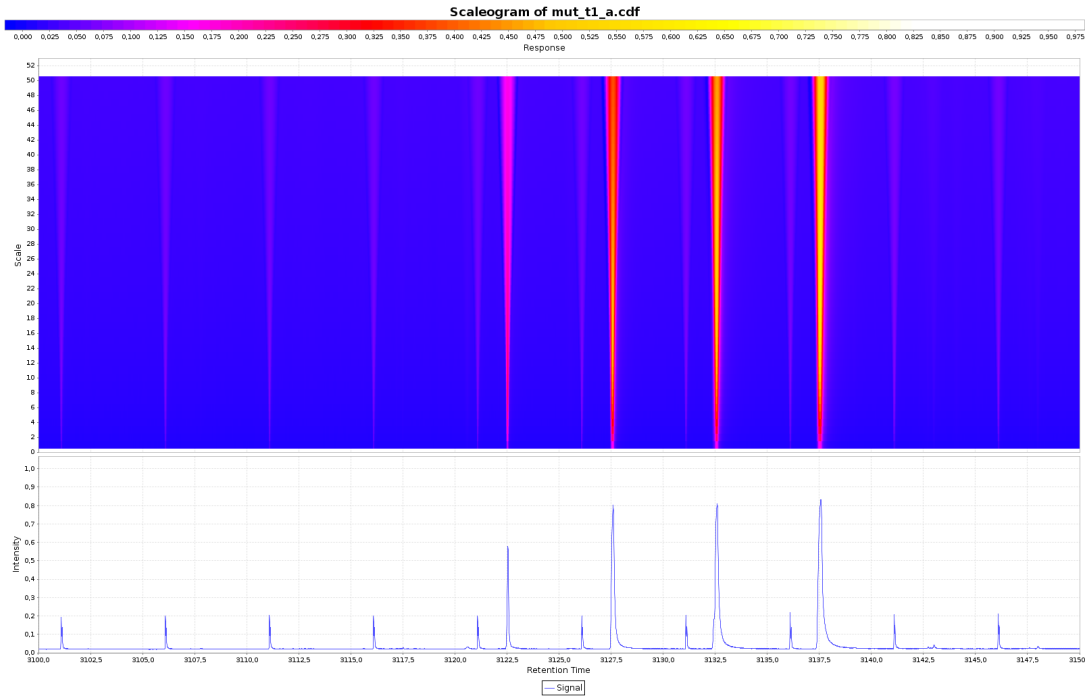


Figure 4.4.: Continuous wavelet transform scaleogram (top) and original signal section (bottom) from a GC×GC-MS chromatogram using the Mexican Hat Mother Wavelet at increasing scales, from bottom to top. Blue values correspond to low response values of wavelet and signal, while orange, yellow and white values represent high response values (good agreement of signal shape and daughter wavelet).

We apply the CWT to the signal $x(\cdot)$ and store the resulting responses for every discrete scale s and discrete translation τ within a two-dimensional matrix with S rows and ℓ columns, equaling the number of scales from minimum scale 1 to maximum scale S and the length of $x(\cdot)$. This matrix is known as the scaleogram of

$x(\cdot)$ with respect to CWT_x^{ψ}. A scaleogram covering 50 scales and approximately 50 seconds of a real GC×GC-MS dataset (see Section 4.4.3 for more details) is shown in Figure 4.4. The color table ranges from blue, for zero response, via red, orange and yellow to white, for the maximal response. It is clearly visible that larger peaks have their maximum response at higher scales. Asymmetric peak shapes prevent the response from attaining higher values.

Computational Complexity of the CWT

The CWT for one (integer) scale can be implemented to run in $\mathcal{O}(\ell)$ time and space (Muñoz, Ertlé, and Unser 2002), where ℓ is the length of the input signal. Thus, for S integer scales, $S \ll \ell$, the CWT requires $\mathcal{O}(S\ell)$ time and space. If the maximum scale is equal to the length of the signal ℓ, the CWT requires at most $\mathcal{O}(\ell^2)$ time and space.

Peak Seed Finding in Scale Space

We follow the idea of Du, Kibbe, and Lin (2006) and locate peaks by following the ridge maxima in the scaleogram, starting at the local response maxima at scale 1, up to a user-defined maximal scale. The method follows the maximum local gradient, allowing a ridge starting from (τ, s) to be extended only into directly adjacent fields at the next higher scale, $(\tau \pm 1, s+1)$, $(\tau, s+1)$, if the response value in any of these candidate fields is larger than the current one. This is only possible because the CWT is highly redundant with respect to the transformation of the original signal into scale space. Since we use a scale progression with unit integer differences, the maximum difference in ridge positions is limited to at most one. Figure 4.5 shows the ridges recovered by the algorithm for a section covering one modulation period (the complete second column retention time period) of a GC×GC-MS TIC. The 20 scales depicted in the figure increase from bottom, just above the colored TIC section, to top. The ridges end at their highest response value. It is visible that the ridges of closely co-eluting peaks recombine at lower scales and that faint peaks tend to have ridges with a larger curvature, when compared to strong peaks. One ridge exceeds the visualized range of scales and thus depicts a peak with wide peak shape (not immediately visible in figure due to coloring).

Dyadic decomposition schemes (sampling the signal at decreasing powers of 2 for faster computation of the transform), such as that used in the discrete wavelet transform (DWT) (Sweldens 1998), do not have a close relationship of the scale space representation of the signal at neighboring scales. Thus, ridges are not easily traceable from low to high scales, as the difference between adjacent ridge elements can be as large as half the sampling period. Additionally, ridges also combine with their neighbors at higher scales, due to the decreasing sampling resolution. As a consequence, it is not immediately possible to determine the optimal response scale of a peak in the signal when the DWT is used.

The maximum response on a ridge corresponds to the optimal scale of the daughter wavelet for the peak originating at the ridge's root at scale 1. A user-definable

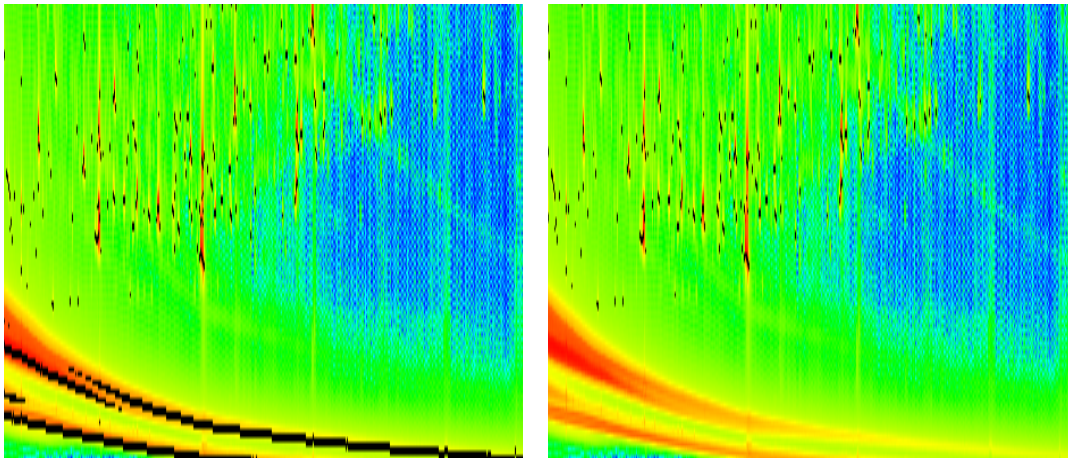
Figure 4.5.: Ridges in scaleogram of GC×GC-MS TIC modulation section covering 20 scales. Ridge seeds are superimposed in black on the original TIC at the bottom. Ridges end at the scale with highest response value.

minimum scale allows to prune peaks below the given scale, thereby removing peaks originating from a noisy background signal (see blue and green checkerboard noise in right hand side areas of Figure 4.6, (a) and (b)). Higher scales can be excluded as they will mostly represent slow baseline changes, thus the maximum scale should be set larger than the expected width of the widest peak in the chromatogram. The minimum scale should be set to the expected width of the narrowest peak in the chromatogram. Figure 4.6 (a) shows the result of the peak seed finding visualized on the original two dimensional TIC. The intensities are color encoded from blue (low intensity), via green (medium intensity) to yellow and red (high intensity). The modulation peak area, an artifact introduced by the cryo-modulator of the LECO Pegasus IV GC×GC-MS (LECO Corp., St. Joseph, MI, USA) is visible in the lower left corner of the chromatogram. Many peaks have been found within that area, indicated by black squares. However, it is also visible that many peaks have been detected within the rest of the chromatogram as well. In Section 4.2.3, we will introduce a method to classify and filter non-informative peaks within the modulation peak areas.

Computational Complexity of the Peak Seed Finding The initial peak seeds can be detected in $\mathcal{O}(\ell)$ time using a simple three-point maxima peak location criterion, such that each peak is a local maximum in the scale space representation of the CWT at scale 1. The number of seeds is limited by the number of elements of x, namely ℓ, and each ridge can be extended over at most $S-1$ scales. For each extension, we need to check whether any of the neighbors $(\tau \pm 1, s+1)$, $(\tau, s+1)$ is larger than the response value at the current position (τ, s). This can be done in constant time for each comparison. In total, for ℓ ridges, we thus have S extensions and therefore a worst-case runtime of $\mathcal{O}(S\ell)$.

4.2.3. Classifying and Removing Uninformative Peaks

We showed how to locate prospective peak seeds in the previous section. Now, we are interested in classifying them based on the density of their two-dimensional neighborhood. The general idea of the approach is that peaks that are closely clustered and thus have a high number of neighboring peaks contained within a fixed radius around them are less informative and more likely to be the result of artifacts in the chromatogram. Such peaks are visualized in Figures 4.6 (a) and (b), superimposed over the modulation ridge that appears as a large red double band extending from the left to the bottom right of the images. Similar patterns appear throughout the two-dimensional chromatogram, but are only faintly visible. In order to support a neighborhood-based classification, we therefore need a data structure that efficiently supports queries of the neighborhood of each peak seed in a given radius.



(a) Before application of the neighborhood filter. Black spots mark the apex positions of peaks in the chromatogram.

(b) Peaks within the lower left modulation ridge have been automatically discarded by the neighborhood filter with $r = 10$ and $\rho = 15$.

Figure 4.6.: Detailed view of peak positions marked as black dots in a GC×GC-MS TIC.

A quadtree is a recursive, spatial indexing data structure for two-dimensional data (Berg 2000; Finkel and Bentley 1974). It decomposes a rectangular region into four regions of equal size (NW, SW, NE, SE) and with constant aspect ratio. The decomposition can be repeated for each new region until no further decomposition is possible (e.g. for discrete images, until each quadrant corresponds to a pixel in the image), or the maximum allowed depth of the tree is reached. For our application, the quadtree's elements will be points in $\mathbb{R}^2$, corresponding to the first and second column retention times of each peak. A point on the two-dimensional plane is added to the quadtree by locating the quadrant that contains its coordinates. A quadrant is split into four new quadrants upon insertion of a new point if its capacity threshold is exceeded. This threshold (called bucketing by Samet (2006)) allows for a

trade-off between search speed and memory efficiency in unbalanced quadtrees and limits the maximum depth of the quadtree. If a point is removed from the quadtree, thereby reducing the fill level of its containing region (bucket) below the threshold, the containing quadrants are merged consecutively into larger quadrants, until the threshold is met again.

For our approach, we are interested not only in constructing a region quadtree by successive insertion of peak seed points, but we are also interested in finding all neighbors within a given radius r around a query peak q. Samet (2006) characterizes such a quadtree as a bucket PR (point region) quadtree.

Quadtree Construction The initial quadtree is empty and we assume that we know all points that will be inserted into the tree and their convex hull and smallest enclosing rectangle beforehand, e.g. from the CWT peak finder introduced in Section 4.2. The enclosing rectangle is required for the PR quadtree in order to calculate the extents of a quadrant in advance. Since the partitioning is fixed and independent from individual data points, the final shape of the tree does not depend on the insertion order, as is the case for point quadtrees.

When we add a point to the root of the quadtree, we need to determine, which of the four quadrants (NW, SW, NE, SE) contains the point. This involves checking for each quadrant's region $R = ([x_0, x_1), [y_0, y_1))$, whether the point (x, y) is contained in it, such that $x_0 \leq x < x_1$ and $y_0 \leq y < y_1$. If it is contained, the point can be added to the corresponding quadrant. Points that lie on the boundaries of a quadrant region ($x = x_1 \vee y = y_1$) are added to the neighboring quadrant (S, E, or SE) that contains them. If the quadrant exceeds its threshold capacity after insertion of a new point, it is split into four new quadrants that are now one level below their parent. All points that were contained in the parent quadrant are then inserted into the four new child quadrants.

Unbalanced region quadtrees do not have a general upper bound on their depth as a function of the number of points in the tree. Samet (2006) shows that the maximum depth of such a tree depends on the minimum Euclidean distance between points in it and is bounded by the following term:

$$\lceil \log_2((s/d)\sqrt{2}) \rceil \,, \tag{4.9}$$

where s is the side length of a (square) quadrant, and d is the minimum Euclidean distance between neighboring points.

The minimum distance for GC×GC-MS data is determined by the scan rate used for the acquisition of mass spectra, determining the minimum time difference in the second separation dimension retention time, and the modulation time period, determining the time difference in the first separation dimension time. Since the minimum distance is achieved by points that are neighbors in one modulation (identical x values), only the difference in the second dimension retention time remains and the Euclidean distance simplifies to the difference between the y values of the neighbors. This difference is the inverse of the scan rate and is usually in

the range between 0.002 (500 Hz) and 0.2 (50 Hz) seconds. Thus, for a GC×GC-MS chromatogram with 5 seconds modulation time and a total runtime of 3600 seconds, the maximum depth is between 12 (5 seconds) and 22 (3600 seconds) levels. A graphical representation of a bucket point region quadtree based on peak seeds found with the CWT is shown in Figure 4.7.

For the case of bucket quadtrees (also called *compressed*), the maximum depth of the tree can be shown to be $\mathcal{O}(\ell)$ in the worst case, yielding construction in $\mathcal{O}(\ell \log \ell)$ time (Aluru and Sevilgen 1999) with insertion and range query running in $\mathcal{O}(\log \ell)$ time.

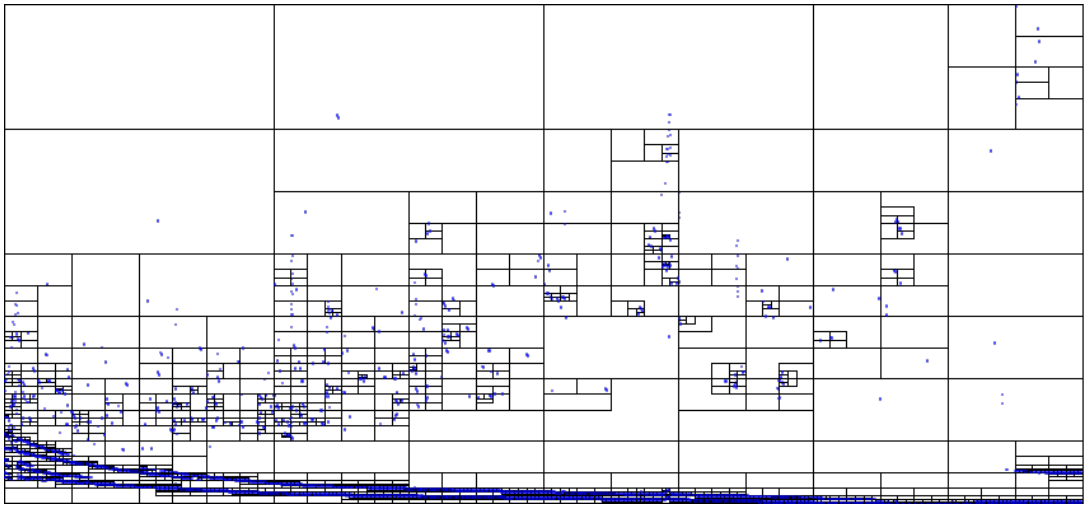


Figure 4.7.: Bucket Point Region Quadtree of the peaks found by the CWT. Quadrants that are not completely subdivided contain fewer peaks than required for the split threshold. The implementation tries to balance the number of quadrants and the number of peaks contained in them for memory efficiency and query speed.

Our implementation follows the ideas of Aluru and Sevilgen (1999) and, thus, Equation 4.9 is an upper bound on the depth of our quadtree implementation.

Eppstein, Goodrich, and Sun (2005) show how a quadtree can be implemented similar to skip lists to achieve a worst case logarithmic depth, which could be an alternative to our implementation for larger datasets where extensive clustering of points in small regions occurs.

Neighbor Search The actual benefit of using the quadtree comes from the efficient search for neighbors within a radius r around a given query point $q(x, y)$. We first need to locate the quadrant region R of q. Starting from the root of the quadtree, we check for each quadrant, whether q is contained in it. If so, we continue to descend into the subtree below until we either find the smallest quadrant containing q or we terminate and return a result indicating that q is not contained in the tree. If we have located q, we need to determine, whether the disc centered at $q(x, y)$ with radius r is fully enclosed within its containing quadrant. This can be done by checking, whether

the enclosing rectangle of the disc is fully contained in the quadrant region R, such that $x_0 \leq x - r$, $x + r < x_1$, $y_0 \leq y - r$ and $y + r < y_1$. If this is the case, one can evaluate the ellipsoid equation describing the disc centered at q and with identical minor and major radius r for every candidate peak in the quadrant:

$$z = \frac{(x_1 - x)^2}{r^2} + \frac{(y_1 - y)^2}{r^2} \,. \tag{4.10}$$

If $z \leq 1$, the candidate point (x_1, y_1) is contained in the query disc and added to the query result list. If the disc overlaps with neighboring quadrants, we add all points from those quadrants that are also contained within the disc's bounds to the query result list. Equation 4.10 is closely related to the elliptical decision criterion used in Equation 4.12 on page 97, which is more general for distinct horizontal and vertical radii.

4.2.4. Neighborhood Density-based Filtering

We apply the range search for every point in the tree for a given radius r and receive the list of neighbors within that radius for each point. The size of the returned list of neighbors for a point indicates whether the point is solitary (few neighbors) or clustered (many neighbors). Figure 4.8 shows the neighborhood density histogram for a radius of 10 s. It is visible, that the neighborhood density distribution is essentially bipartite (if we omit the maximum around 65 neighbors), with the first maximum between 10 and 15 and the second maximum between 40 and 45 neighbors.

Points and their corresponding ridges with a neighborhood density above a user defined threshold are removed from further consideration, in order to avoid the downstream processing of peaks carrying highly redundant analytical information. The automatic selection of the neighborhood density threshold is non-trivial. As of now, it needs to be determined manually by the user. An automation would be possible by analyzing the bipartite neighborhood density distribution for a larger number of different GC×GC-MS datasets and ensuring that this property is generally found in GC×GC-MS data. If this is the case, one could fit a Gaussian mixture model for two Gaussian probability density functions, centered at the maxima of the empirical neighborhood density distribution, and select the neighborhood density value with the fewest points as the decision boundary between the distributions. However, experiments with the neighborhood density threshold have shown that choosing this value leads to a high number of false positive peaks. For a low number of false positives, the boundary needs to be moved towards the first maximum of the neighborhood density distribution, at the risk of losing true positive peaks that happen to have a higher than average neighborhood density.

In the next section, we will show how peaks found by either the algorithm described in the previous sections or by third party software can be compared automatically between different chromatograms. We will briefly explain the peak area integration and fusion method available in MALTCMS in Chapter 5.

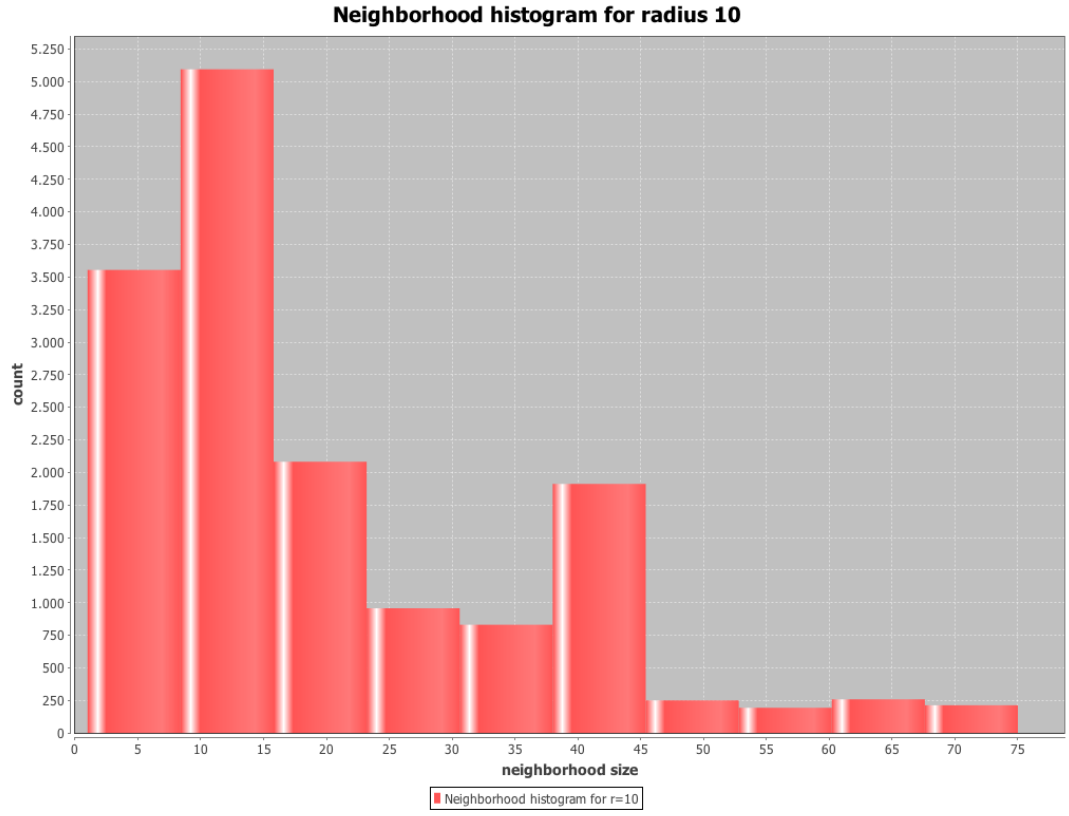


Figure 4.8.: Ridge Neighborhood Histogram for $r = 10$ s. The distribution of the histogram is bipartite. The first maximum lies between 10 and 15 neighbors, while the second maximum (especially modulation area peaks) lies between 40 and 45 neighbors. For Figure 4.6 b), all ridges with more than 15 neighbors were removed.

4.3. Peak Alignment for GC×GC-MS

Preprocessing of GC×GC-MS data involves the filtering and noise reduction of the raw signal, the localization, deconvolution, integration, and normalization of analyte signals of interest (peaks) (Amador-Muñoz and Marriott 2008), as well as downstream matching of peaks to create a multiple alignment of related signals from different samples. We already described the peak alignment problem in Section 2.6.4 and discussed its necessity in the context of GC-MS, LC-MS, and GC×GC-MS. In the following sections, a novel automated method for the multiple alignment of GC×GC-MS peaks is introduced: Bi-directional best-hit peak assignment and clique extension for two dimensional chromatograms (BIPACE 2D), which is based on comparing peak mass spectra and retention times in two dimensions between a large number of samples.

4.3.1. Background

For GC×GC-MS peak alignment, the mass spectrum behind each time point on the two-dimensional chromatographic plane can be used as an additional criterion for peak similarity or identity. The use of EI in GC×GC-MS produces rich fragmentation spectra that are comparable to fingerprints of each analyte. However, EI is known to lead to identical or very similar spectra (due to detector noise) for certain classes of analytes, especially structural isomers. Thus, an additional criterion for their distinction is required, such as the retention time (RT) information of the mass spectrum in the first and second dimensions of separation. A peak in GC×GC-MS may encompass many mass spectra, so that a reduction to a representative mass spectrum is advisable for improved signal-to-noise ratio and better spectral database search results (Oh et al. 2008). If those results are reliable and return few false positive identifications, one can subsequently use the assigned names to associate peaks across samples. However, the results of database searches may consistently associate a spectrum erroneously with an analyte that happens to be just above the identification threshold used by the database, while the true analyte is missing from the database.

Typically, peaks should be aligned between samples that were measured under identical (homogeneous) separation conditions. However, the algorithms of Jeong et al. (2012), Kim, Fang, et al. (2011), and S. Wang et al. (2010) also support alignment of peaks that were measured under different (heterogeneous) conditions (e.g. different temperature gradient) that lead to non-linear shifts especially in the first retention time of the GC×GC chromatogram.

MSORT The MSORT algorithm (Oh et al. 2008) sorts and associates peaks based on their absolute retention time difference for each separation dimension and mass spectral similarity using Pearson's correlation coefficient. It successively builds a sorted peak table created from unassigned peak tables and matches peaks from a reference table, a search table with the highest number of merged peaks, against the remaining peaks using a sorting criterion until all searchable peaks have been processed.

DISCO The algorithm DISCO (Wei et al. 2013; S. Wang et al. 2010) uses landmark peaks in each sample that are mapped to landmark peaks in a reference sample using Euclidean distance to calculate retention time similarity and Pearson's correlation coefficient to determine the similarity of mass spectra. Based on the landmark peaks, the method determines a local linear interpolation that is applied to non-landmark peaks, thereby correcting for non-linear retention time distortion.

MSPA and SWPA Kim, Koo, et al. (2011) and Kim, Fang, et al. (2011) have introduced two different algorithms to approach the peak alignment problem in GC×GC-MS. The SWPA approach uses variants of dynamic programming to find a peak

matching with maximal score for pairwise alignments. Their MSPA method includes the optimization of a likelihood function based on a parametrized mixture similarity, that involves the dot product as mass spectral similarity and retention time deviation calculation with different distance metrics. Both of their approaches extend the pairwise alignments transitively to a multiple peak alignment, based on a prior chosen reference peak list.

GUINEU We previously described the SCORE ALIGNMENT algorithm used by GUINEU (Castillo et al. 2011) in Section 4.1.1, but repeat it here in greater detail for completeness. SCORE ALIGNMENT is based on a combined score, using pre-defined windows for first and second dimension retention time deviations and RI deviation. The method scores neighboring peaks against potential target peak groups, building candidate paths of related peaks. The weighted cosine product is used to avoid alignment of mass spectra with low pairwise scores, with a user-defined minimum threshold. Path-generation and evaluation is performed in parallel and followed by a subsequent post-processing phase, where peaks that were assigned to multiple groups are reassigned to the peak group with highest score until all such conflicts are resolved. We will refer to the SCORE ALIGNMENT method as GUINEU in the remainder of this chapter.

MBPA Jeong et al. (2012) use a statistical model to align the peaks, based on pairwise peak scores calculated from mass spectral similarity (cosine score) and retention time deviation score functions. Their approach uses landmark peaks with a high posterior probability according to their model to calculate a retention time correction for the remaining peaks which fall below a specific posterior probability threshold. They additionally calculate a corrected retention time for aligned peaks.

CCM Reichenbach et al. (2013) use the consistent cliques method (CCM), an approach that is in principle similar to the BIPACE method already described in Hoffmann et al. (2012), but for GC×GC-MS data where pairwise matches between multiple peak lists have already been determined. However, their algorithm can only merge conflict-free cliques above a user-defined threshold and thus may report fewer cliques than BIPACE. CCM is part of the commercial software GC Image (Lincoln, NE, USA).

BIPACE and BIPACE RT The BIPACE and BIPACE RT algorithms have been described in Section 3.2, showing their applicability for peak alignment of GC-MS data. BIPACE 2D is a novel extension of BIPACE that uses the two-dimensional retention time information in addition to mass spectral similarity to align peaks across multiple chromatograms without requiring a user-defined reference chromatogram.

4.3.2. BIPACE 2D Pairwise Peak Similarity Function

We extend our notation from Section 3.3 for the two-dimensional retention time domain in GC×GC-MS. Given a chromatogram $C = \{p_1, p_2, ..., p_\ell\}$ as an ordered set of peaks, we define a two-dimensional peak $p = (\mathbf{m}, \mathbf{i}, t_1, t_2)$ as a quadruple of a mass vector $\mathbf{m}$, an intensity vector $\mathbf{i}$, both with the same dimensions, a first column retention time t_1, and a second column retention time t_2. For two peaks p and q, represented by their binned mass spectral intensity vectors with first column retention times $t_{1,p}$, $t_{1,q}$, second column retention times $t_{2,p}$, $t_{2,q}$, and retention time tolerances of D_1 and D_2, for the first and second column, respectively, we define a similarity function following Robinson et al. (2007) as:

$$f_{2d}(p,q) := \exp\left(-\frac{(t_{1,p}-t_{1,q})^2}{2D_1^2}\right) \cdot \exp\left(-\frac{(t_{2,p}-t_{2,q})^2}{2D_2^2}\right) \cdot s(p,q)\ , \tag{4.11}$$

where $s(p,q)$ is an arbitrary similarity function between the mass spectral intensity vectors, such as the cosine, the weighted cosine (Stein and Scott 1994), the dot product, Pearson's linear correlation coefficient, or Spearman's rank correlation coefficient. $f_{2d}(\cdot,\cdot)$ can be interpreted as a likelihood function that independently scores the proximity and mass spectral similarity of its arguments. It is maximized by peaks that have very low deviation in retention times and a very high mass spectral score. The impact of deviations in either retention time dimension can be individually adjusted via the retention time tolerance parameters D_1 and D_2 of the Gaussian RT penalty terms, where a higher value allows for larger retention time deviations.

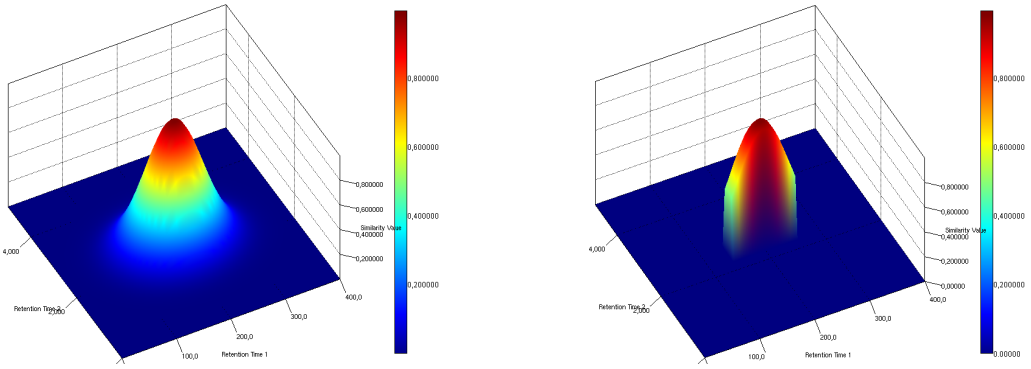


Figure 4.9.: Product of Gaussian retention time penalty functions with $D1 = 50$, $D2 = 0.5$. Left: without thresholds; Right: with thresholds $T1 = 0.5$, $T2 = 0.9$.

To prune the search space early during the pairwise all-against-all peak similarity calculation phase of our algorithm, each RT penalty term has an additional threshold parameter (T_1 and T_2, respectively) that allows to effectively stop any further evaluation of $f_{2d}(\cdot,\cdot)$ if the value of the threshold for that term is not attained or exceeded. Thus, the mass spectral score function may not need to be evaluated at all, resulting in a large speedup at the expense of reduced sensitivity towards peaks with larger

retention time deviations. Figure 4.9 shows the three-dimensional surface as defined by Equation 4.11 for typical retention time tolerances, without and with threshold application. The value of $s(p, q)$ was fixed to 1 for these plots.

The BIPACE 2D algorithm is essentially identical to BIPACE after the pairwise similarity calculation has been performed. The essential background of the clique finding and merging phases together with runtime and space complexity results is presented greater detail in Section 3.3.

Following our results in Section 3.3.3, the time and space complexity of BIPACE 2D and BIPACE are equivalent to $\mathcal{O}(K^2\ell^2)$ in time and $\mathcal{O}(K^2\ell)$ in space, where K is the number of chromatograms and ℓ is the upper bound of the number of peaks in each chromatogram.

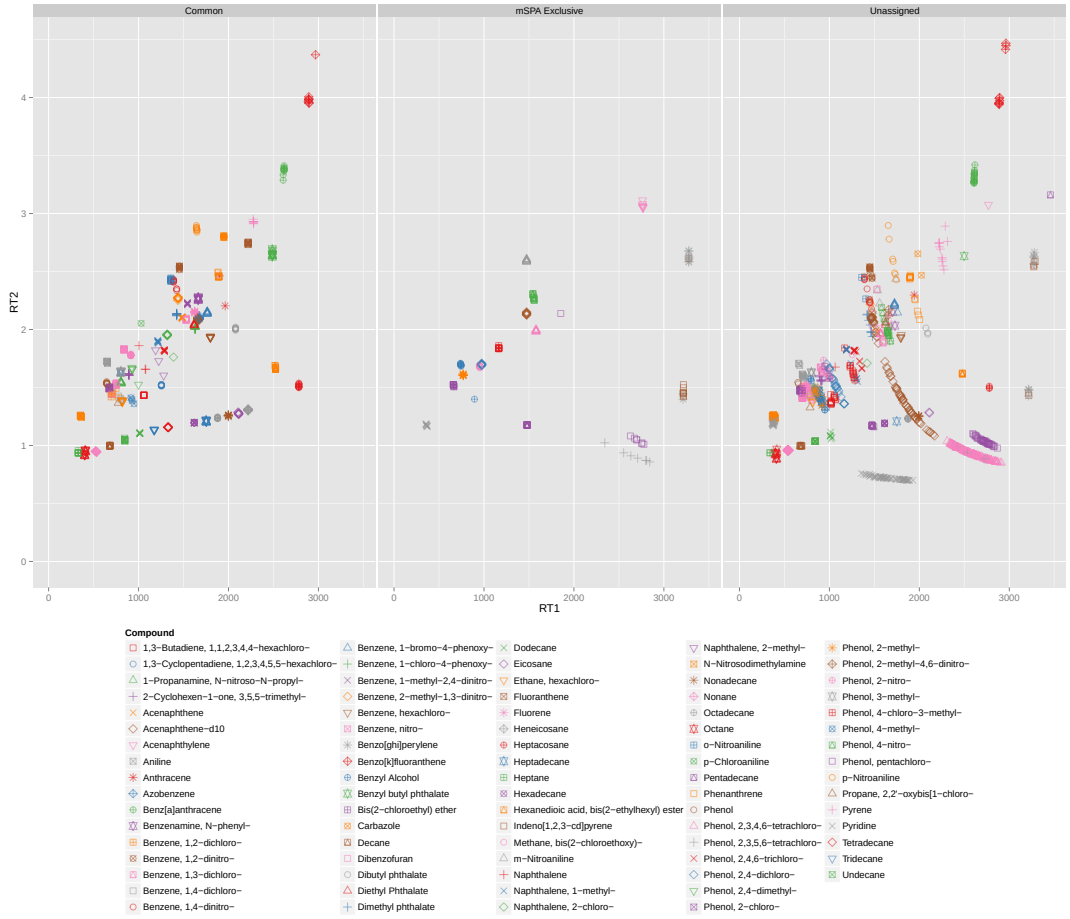


Figure 4.10.: Peak set partitions for MSPA dataset I, showing peaks common two GMA and MGMA in the leftmost tableau, peaks exclusive to GMA (MSPA) in the middle, and peaks that were unassigned by either method in the rightmost tableau. The peak set created by MGMA is a perfect subset of the GMA peak set. Thus, there are no peaks that are exclusive to MGMA.

Table 4.3.: Parameters used for alignment reference generation for the different datasets. a is the major and b the minor radius of the elliptical decision boundary function z (Equation 4.12).

			# of peaks in reference		
Dataset	a	b	GMA	MGMA	manual
mSPA I	50.0	0.5	752	592	–
mSPA II	55.0	0.5	1682	1081	–
SWPA I	800.0	0.5	1201	1090	–
chlamy I	250.0	0.5	2723	1629	436

4.3.3. Reference Dataset Generation

In order to evaluate their two-dimensional retention time alignment algorithm mSPA, Kim, Fang, et al. (2011) use the raw peak lists as created by the ChromaTOF software (LECO Corp, St. Joseph, MI, USA) and create reference multiple alignments based on the assigned peak names. Since each peak list can contain multiple peaks with the same name, the authors employ a method to resolve such potential conflicts by selecting the peak with the largest recorded area as the representative for a group of otherwise identically named peaks. This approach is referenced in the remainder of this chapter as *grouping by maximum area* (GMA). As we have investigated, GMA may lead to arbitrary and spurious assignments of peaks to the same alignment row (with identical names across peak reports), hampering the clear definition of what *true* and *false* positives as well as negatives are. Examples of potentially problematic assignments for mSPA Dataset I (see Section 4.4.1 for details) are the compounds (*Naphthalene*), (*Naphthalene, 2-methyl-*), (*Anthracene*), (*Benzo[ghi]perylene*), (*Indeno[1,2,3-cd]pyrene*), (*Phenol, pentachloro-*), and (*Phenol, 2,3,5,6-tetrachloro-*), all of which appear to have been assigned the wrong name in a number of cases, as is shown in Figure 4.10. Corresponding plots for the other datasets are available in Section C. The complete data comparing the GMA reference creation approach to our proposed approach for all datasets used in this work are available online[3].

The MGMA approach

In order to address the issues with the approach used by Kim, Fang, et al. (2011), the reference generation method was modified to remove spurious assignments that relate back to potentially false assignments of peak names by the ChromaTOF software. As mentioned before, additional problems may arise from the selection process that relies solely on picking the peak with the largest area as the representative for a group of identically named peaks within each report.

3. http://maltcms.sf.net/pub/bipace2d

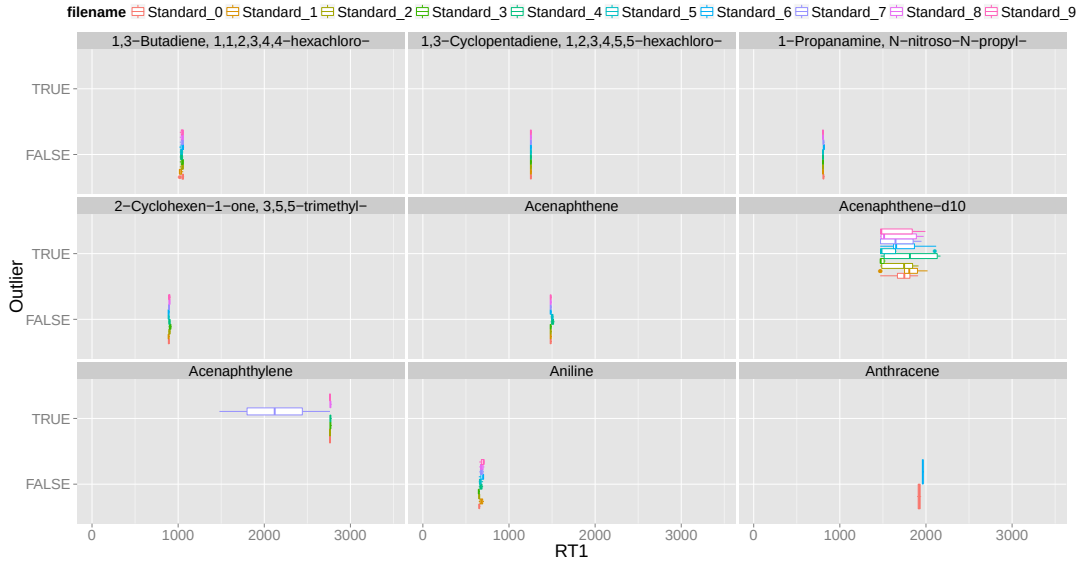


Figure 4.11.: Box plots of the first column retention time for a subset of peaks from mSPA dataset I. Peak groups that exceed the standard deviation thresholds are marked as suspicious, indicated as *TRUE*, i.e. *Acenaphtylene* in sample Standard_7 (range of 1200 s), while peak groups within the decision boundary (see Eqn. 4.12), are indicated as *FALSE*. The distinction by originating sample name shows that there are groups that appear to have correct peak name assignments in some of their samples, as indicated by the low deviation in retention time, but that have largely varying retention time deviation in other samples.

The new method *modified grouping by maximum area* (MGMA) calculates, for all equally named peaks (peak groups) in all peak reports, the standard deviation of the retention times in the first and second dimension of separation.

For an arbitrary group of peaks P with the same name, $x := \sigma\big(t_1(P)\big)$ and $y := \sigma\big(t_2(P)\big)$ are defined as the standard deviations of the peak group retention times in the first (t_1) and second (t_2) dimension of separation. An elliptical function centered at $x_0 = 0$ and $y_0 = 0$ and with a major radius a (maximum allowed standard deviation for t_1) and minor radius b (maximum allowed standard deviation for t_2) is then used to calculate the decision criterion z:

$$z = \frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} \,. \tag{4.12}$$

If $z \leq 1$, the group is retained. Otherwise, if $z > 1$, the group at (x, y) is outside of the bounds of the ellipse defined by x_0, y_0, a, b and is marked as a potential outlier group. The parameters used for a and b for the different datasets examined in Section 4.4 are given in Table 4.3.

Figures 4.11 and 4.12 show the retention time deviation behavior of a selection of peak groups within mSPA Dataset I for the first and second column retention times,

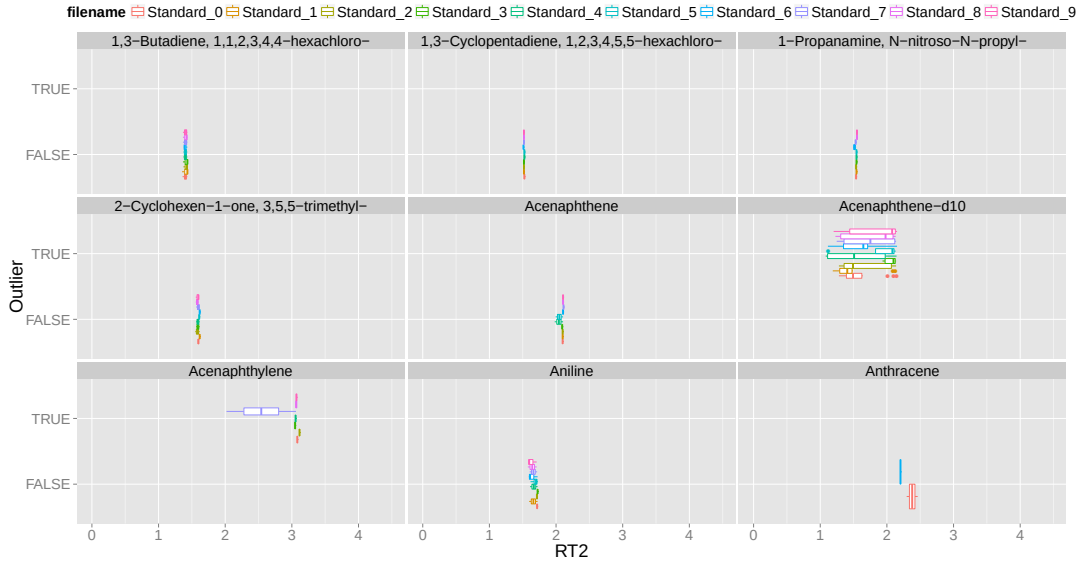


Figure 4.12.: Box plots of the second column retention time for a subset of peaks from mSPA dataset I. Peak groups that exceed the standard deviation thresholds are marked as suspicious, indicated as *TRUE*, i.e. *Acenaphtylene* in sample Standard_7 (range of 2 s), while peak groups within the decision boundary (see Eqn. 4.12), are indicated as *FALSE*. As in Figure 4.11, the distinction by originating sample name shows that there are groups that appear to have correct peak name assignments in some of their samples, while the retention time deviation is large within the other samples.

together with their suspected outlier behavior according to the decision boundary (*TRUE* for suspected outlier, *FALSE* otherwise). The visualizations show, for each peak group with identical putative name, the retention time behavior separated by the source file of the peaks. This clarifies that ChromaTOF does indeed export multiple peaks per file with the same putative name, some of them seemingly false identifications, thereby severely distorting the peak groups runtime deviation. However, there is no apparent trend among the input files to produce more or fewer outliers than the other files, so the observed effect may be attributed to random misannotations and not to a systematic error. It is notable that the retention time deviation on the second separation dimension exceeds one second, which is rather atypical for the usually only slightly varying second dimension retention time over multiple modulations (Vial et al. 2009) and between samples (under the same analytical conditions).

The decision boundaries (shaded dark grey) along with the standard deviations of the group-wise retention times for a selection of peaks from mSPA Dataset I are shown in Figure 4.13. Additionally, the figure visualizes the result of the decision criterion. Potential outlier groups are indicated by a red color, groups that are within the defined bounds are indicated by a blue color. Most outlier groups have a very large deviation in the first retention time dimensions, which supports the claim that

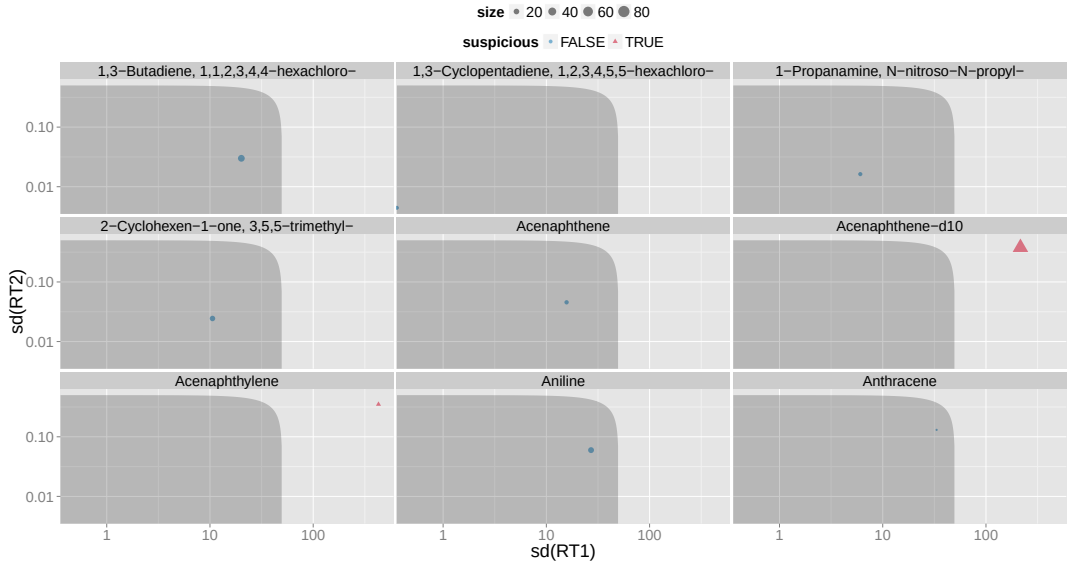


Figure 4.13.: Within-group standard deviations of peak retention times on the first and second separation column for a subset of peaks from mSPA dataset I. Peak groups that exceed the standard deviation thresholds (see Eqn. 4.12) are marked as suspicious, before being removed from the reference alignment. The size of the glyph represents the number of peaks contained in a compound group, while colour and shape encode whether the group is suspicious (red triangles) or not (blue circles).

member peaks of those groups have been erroneously assigned the same putative name.

Peaks belonging to outlier groups are removed by MGMA without further consideration, since a large deviation in one or both retention time dimensions may be a strong hint towards wrongly assigned peak names. An approach to further discriminate the members of such groups may lead to additional sources of false peak assignments and has thus not been considered at this stage.

Additionally, all peaks that occur only once throughout all peak reports are removed, since they can not provide any reliable grouping information and may again have resulted from spurious identifications by the vendor software due to different sample quality and/or non-optimal parameter settings used during peak detection and putative peak identification.

The final reference multiple alignment is then created using GMA on the remaining peaks. MGMA thus reports a completely contained subset of the original peaks as reported by GMA. An example for this is given in Section 4.4.3.

4.3.4. Peak Alignment Performance Evaluation

A reference alignment peak group defines whether a peak, represented by its index in the original peak list, is present in a sample or absent. Each column in the reference

alignment corresponds to one sample's peak list, while each row represents an aligned peak group, spanning multiple samples. The results of each alignment algorithm are tested against each reference alignment group until either a match is found or the group is reported to be nonassignable to a counterpart in the reference alignment.

In order to evaluate every parameterization of each alignment algorithm, we use the same measures that we already described in Section 3.5, namely precision, recall and F1 score, based on true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). For each multiple alignment result obtained from an algorithm parameterization, all unmatched peaks of the reference alignment, excluding absent ones, are added to the number of false negatives to normalize Recall and F1 score with respect to the size of the reference alignment.

We additionally evaluate the alignment performance of each algorithm based on a comparison of the average performance $F1_p$ over all pairwise alignments, following the evaluation method reported by (Kim, Fang, et al. 2011; Kim, Koo, et al. 2011).

We discuss the advantages and disadvantages of our row-wise multiple alignment evaluation over the pairwise alignment evaluation as used by (Kim, Fang, et al. 2011; Kim, Koo, et al. 2011), in Appendix C.7. Differences in the resulting numbers for $F1_p$ to the numbers published in Kim, Koo, et al. (2011) are due to the evaluation scheme that we use. We compare all possible unique pairwise combinations of alignment column pairs against the corresponding reference alignment columns. Thus our absolute TP, FP, TN, and FN numbers are higher and the $F1_p$ tends to be generally lower.

4.4. Results and Discussion

The GNU R scripts available with Kim, Koo, et al. (2011) and Kim, Fang, et al. (2011) were carefully adapted, in order to be able to run them within an automated evaluation pipeline. The mSPA and SWPA data set peak lists were used unaltered as input to all evaluated programs, keeping peaks that were split across multiple modulations, while removing peak artifacts with an identical area. In order to make the gap-less multiple alignment output of mSPA and SWPA comparable to the gapped multiple alignment of BiPACE 2D, we modified the corresponding R-code to not remove incomplete peak groups. Guineu was modified to parse the ChromaTOF peak file format with separate fields for first and second column retention times and was further adapted to run without a graphical user interface and to record the original row index of each peak in the original peak list for later evaluation.

The algorithms BiPACE, BiPACE RT, and BiPACE 2D were evaluated against Guineu's score alignment (Castillo et al. 2011), mSPA (Kim, Koo, et al. 2011) and its variants PAD, PAS, DW-PAS, SW-PAD, and PAM, as well as against SWPA (Kim, Fang, et al. 2011) and its variants SWRM, SWRE, SWRME, and SWRME2. The mSPA, SWPA, and Guineu methods used the ChromaTOF peak lists as input directly. For the BiPACE methods, we converted the ChromaTOF peak lists to a backwards compatible, extended netCDF format (Rew and Davis 1990), supporting first

and second column elution time. BIPACE also supports mzML input files (Martens et al. 2011) containing the standardized spectrum attributes *first_column_elution_time* and *second_column_elution_time*. The paramterizations reported as optimal by both the OP-PAM and the likelihood-based parameter optimization for the SWPA methods were explicitly included in the evaluation for each of the respective methods. The results for the MSPA SW-PAD variant using Pearson's correlation between spectra and Euclidean distance for retention time matching correspond to the results of the DISCO algorithm (Kim, Koo, et al. 2011). A detailed overview of the best results for each dataset and variant is available in tabular form in Appendix C.

Each algorithm was run and evaluated for a range of different parameter values. The user-configurable parameters (penalty terms, mass spectral score function, retention time distance function) for MSPA and SWPA were taken from the corresponding publications (Kim, Koo, et al. 2011; Kim, Fang, et al. 2011). We tested all viable combinations of score function (dot product, linear correlation) and retention time distance functions (Manhattan, Euclidean, Canberra, Maximum). Kim and Zhang (2013) provide a recent comparison of MSPA using an additional set of similarity functions that were not evaluated here. For BIPACE and its variants, the varied parameters included the mass spectral score function, retention time penalty terms (BIPACE RT and BIPACE 2D), and retention time penalty threshold (BIPACE RT and BIPACE 2D). The parameter values for all methods are available for each dataset individually within Supplementary File 2[4]. Plots of the runtime and memory usage of each parametrized method are available in Appendix C. They reflect only the peak alignment phase, not the data import and filtering phases of the algorithms.

4.4.1. MSPA Datasets

The authors of the MSPA publication (Kim, Koo, et al. 2011) evaluated their algorithms on two different datasets. The first one, here termed MSPA dataset I, consists of ten samples of 106 standard compound mixtures, measured throughout with the same temperature gradient. It contains 1672 peaks in total, of which 752 in 81 rows were used in the GMA reference alignment. These were further reduced to 592 peaks in 64 rows by MGMA.

The second dataset, MSPA dataset II, contains five samples of rat plasma with spiked-in 6–compound standards, also measured under identical temperature gradient conditions. The original peak reports contained 3575 peaks. These were reduced to 1682 peaks in 493 rows by GMA's reference alignment generation, and further reduced by MGMA to 1081 peaks in 320 rows.

Table 4.3 holds the parameters used to generate the MGMA reference alignments, for MSPA dataset I and MSPA dataset II, respectively.

Figure 4.14.: F1 score for all parameterizations of the evaluated algorithms for mSPA dataset I. BiPACE 2D using the weighted cosine as similarity function between mass spectra outperforms all other methods.

Results for mSPA Dataset I

Figure 4.14 shows the F1 scores obtained for each of the methods under consideration, against both GMA and MGMA reference alignments. For both references, BiPACE 2D achieves the highest F1 scores (0.9296 for GMA and 0.9472 for MGMA reference), followed by BiPACE RT (0.9181 and 0.9322). The Guineu (0.9082 and 0.9165) and mSPA-PAM (0.905 and 0.9169) variants follow closely behind. BiPACE 2D also has a consistently better Precision value (0.9551 and 0.9607) than any of the other methods. On both references, the best BiPACE 2D instance uses retention time penalty parameters of $D_1 = 10$ s, $D_2 = 0.5$ s, $MCS = 2$, the dot product as mass spectral similarity, and retention time penalty thresholds of $T_1 = 0$, and $T_2 = 0.99$, effectively allowing only very small differences in the second dimension retention time. BiPACE 2D also achieves the best average pairwise $F1_p$ scores, 0.9203 ± 0.022 with $D_1 = 10$ s, $D_2 = 0.25$ s, $MCS = 2$, $T_1 = 0$, $T_2 = 0.25$, and 0.9374 ± 0.021 with $D_1 = 10$ s, $D_2 = 0.5$ s, $MCS = 2$, $T_1 = 0$, $T_2 = 0.99$, each time using the dot product as mass spectral similarity. More details may be found in Appendix C.2.

4. Available at http://maltcms.sf.net/pub/bipace2d

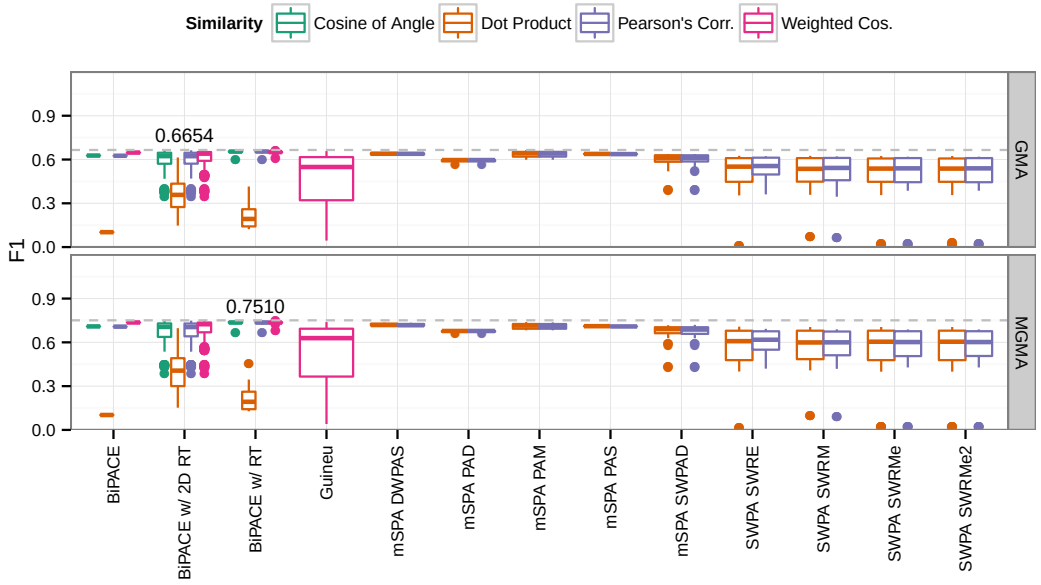


Figure 4.15.: F1 score for all parameterizations of the evaluated algorithms for mSPA dataset II. BiPACE 2D and BiPACE RT perform better than any other method using the weighted cosine and Pearson's linear correlation as similarity functions between mass spectra.

Results for mSPA Dataset II

Comparing the F1 scores for mSPA dataset II, BiPACE 2D (0.6654 on GMA reference) and BiPACE RT (0.751 on MGMA with $D = 25$ s, $T = 0.25$) perform better than any Guineu, mSPA, or SWPA variant (see Figure 4.15). The best instances use the weighted cosine or cosine mass spectral score function, or Pearson's linear correlation, and not the dot product, in comparison to the results in mSPA dataset I, where the dot product was more competitive. The best BiPACE 2D instance on the GMA reference also achieves the highest $F1_p$ value (0.6857 ± 0.0264) with parameters $D_1 = 25$ s, $D_2 = 0.5$ s, $MCS = 2$, and retention time penalty thresholds of $T_1 = 0.75$, and $T_2 = 0$, effectively allowing only very small differences in the first dimension retention time, while allowing larger differences in the second dimension retention time. BiPACE RT scores the highest $F1_p$ value of 0.8231 ± 0.0201 on the MGMA reference with $D = 30$ s, $T = 0.9$ and $MCS = 2$. The F1 and $F1_p$ values for Guineu, mSPA and the SWPA variants do not fall far behind in this case on either reference alignment in comparison to the other datasets (see Appendix C.3 for more details).

4.4.2. SWPA Dataset

The SWPA publication (Kim, Fang, et al. 2011) used two different datasets for evaluation purposes. However, SWPA dataset II (spiked-in) was excluded from this evaluation because it was identical to mSPA dataset II. SWPA dataset I is a combination of 16 samples, measured using three different temperature gradients. It should therefor be a significant challenge for retention time-based algorithms. The dataset originally contained 2499 peaks, which were reduced to 1201 peaks in 83 alignment rows by GMA, and to 1090 peaks in 75 rows by MGMA. The parameters for the MGMA reference alignment are given in Table 4.3 on page 96.

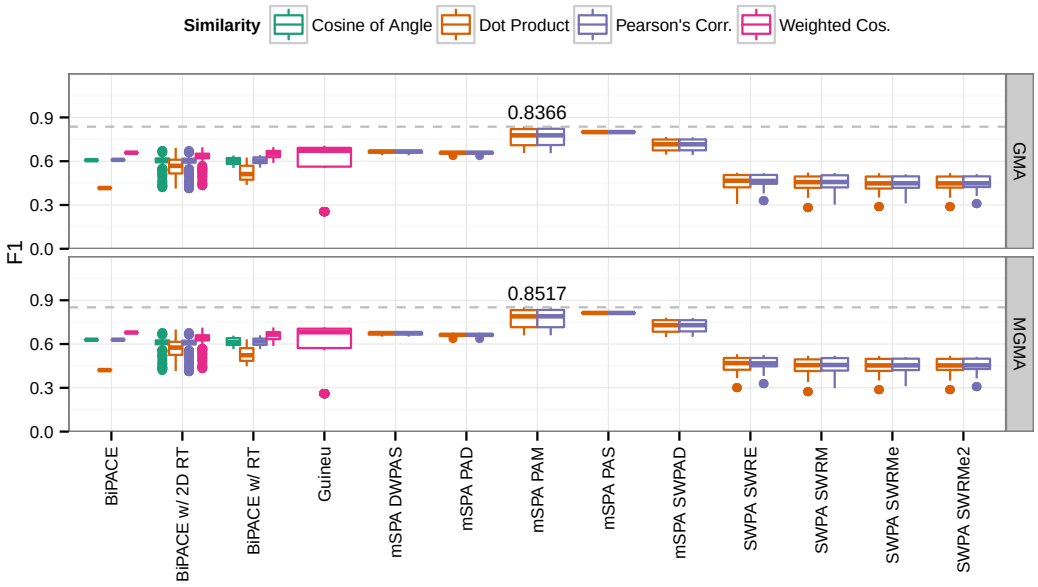


Figure 4.16.: F1 score for all parameterizations of the evaluated algorithms for SWPA dataset I. mSPA-PAM, mSPA-PAS, and mSPA-SW-PAD perform better than the BiPACE, SWPA, and Guineu variants.

Results for SWPA Dataset I

For this dataset, the mSPA-PAM variant using dot product as pairwise spectral similarity and maximum distance for retention time difference performs best when considering F1 score, with values of 0.8366 (GMA reference) and 0.8517 (MGMA reference). The best Guineu instance achieves values of 0.7061 and 0.7148, while the best BiPACE instance achieves F1 values of 0.6966 and 0.7136, respectively. More details are shown in Figure 4.16. Considering the $F1_p$ score, the order is unchanged, with mSPA-PAM achieving 0.7942 ± 0.0914 and 0.8101 ± 0.0882, Guineu scoring

0.5142 ± 0.314 and 0.529 ± 0.309, and BIPACE trailing with 0.4772 ± 0.248 and 0.4995 ± 0.2479, on GMA and MGMA references.

The comparatively low F1 score for all BIPACE variants can be explained by many peaks that are reported as being absent, while they are present in either reference alignment. Those absent peaks are counted as FNs and thus lead to a low Recall value (see Appendix C.4). However, the BIPACE variants perform better when considering TN and FP values. They report fewer peaks per peak group, resulting in a more conservative alignment when compared to either the MSPA or SWPA variants, at the expense of more TPs.

BIPACE achieves high Precision values for either reference (0.9049 and 0.9146), but lacks in Recall (0.5174 and 0.5397), leading to the comparatively low F1 and $F1_p$ scores, while BIPACE 2D has slightly lower Precision values (0.8588 and 0.8631) but higher Recall (0.5841 and 0.607). Detailed numbers for the best instances are given in Table C.3 on page 226. A more detailed table including individual parametrizations of the best instances is available online[5].

4.4.3. *Chlamydomonas reinhardtii* Dataset

The *Chlamydomonas reinhardtii* dataset (CHLAMY dataset I) was originally analyzed in Doebbe et al. (2010). The experiment explored the difference in H_2 production yield between the *C. reinhardtii* wild type strain *cc406* (WT) and the high H_2-producing strain *Stm6Glc4* (MUT) at two different time points, namely before (T1) and during (T2) the H_2 production phase, with three replicates for each of the factor combinations WT-T1, WT-T2, MUT-T1, MUT-T2, yielding a total of 12 samples. The stored original samples of that experiment were prepared according to the protocol in Doebbe et al. (2010) and then reanalyzed using a LECO Pegasus 4D time-of-flight mass spectrometer (LECO, St. Joseph, MI, USA). The Pegasus 4D system was equipped with an Agilent 6890 gas chromatograph (Agilent, Santa Clara, CA, USA).

Sample Acquisition

Splitless injection of 1 μl sample volume was conducted at 275°C injector temperature. The gas chromatograph was equipped with a 30 m x 0.25 mm x 0.25 μm film thickness, Rtx-5ms (Restek Corp., Bellefonte, PA, USA) capillary column used as the primary column and a BPX-50 (SGE Incorporated, Austin, TX, USA) 2 m x 0.1 mm x 0.1 μm capillary column used as the secondary column. The temperature program of the primary oven was set to the following conditions: 70 °C for 2 min, 4 °C/min to 180 °C, 2 °C/min to 230 °C, 4 °C/min to 325 °C hold 3 min. The temperature program of the secondary oven was set with an offset of 15 °C to the primary oven temperature. The thermal modulator was set 30 °C relative to the primary oven and used a modulation time of 5 s with a hot pulse time of 0.4 s. The mass spectrometer ion source temperature was set to 200 °C and the ionization was performed at -70

5. http://maltcms.sf.net/pub/bipace2d

eV. The detector voltage was set to 1600 V and mass spectra were recorded at 200 scans/second using a scanning range of 50-750 m/z.

Sample Processing

The samples were processed automatically by the LECO ChromaTOF software v.4.22 at a signal to noise (S/N) ratio of 100. The baseline offset was 0.8 and the two peak widths were set to 0.2 s (as measured from baseline to baseline) and 15 s (first dimension). By using the classification feature of the software, background peaks originating from column bleed or solvent tailing were removed.

Analytes were putatively identified by database searches using the GMD, version 20100614 (Hummel et al. 2007). The minimum required similarity threshold for assignment of a compound name was set to 600 on a scale from 0 for no similarity to 1000 for identity. The original ChromaTOF peak lists contained a total of 31695 peaks for the 12 samples. All peaks with best matching library spectrum similarity below 600, flagged as 'Unknown', were removed from further consideration. The original peak lists were exported from ChromaTOF using one field ('R.T. (s)') for the first and second column elution time. We therefor introduced two separate columns for first and second column elution time ('1st Dimension Time (s)' and '2nd Dimension Time (s)') to make them suitable as input to both MSPA and SWPA.

The resulting peak lists for each sample were further rectified by removing all peaks with unclear GMD identifications containing an 'NA'. These steps were required in order to make the peak lists compatible to MSPA's and SWPA's peak merging preprocessing step, which was needed for the generation of the evaluation reference alignments with GMA and MGMA. These unknowns would otherwise have lead to false peak group assignments based on the peaks' non-unique names. The removal of 'Unknown' peaks and rectification of 'NA's reduced the number of peaks to a total of 4860. The final GMA reference alignment contained 2723 peaks in 369 rows, while the MGMA reference, using the parameters given in Table 4.3, contained 1629 peaks in 224 rows.

Manual Reference

To define the manual reference alignment, the reduced peak lists without 'Unknown's and 'NA's were inspected and only peaks were kept that could be positively confirmed by assigned name and retention times within two of the three samples within each factor combination of the experiment. For a number of unclear cases, we additionally compared the mass spectra of the questionable peaks manually to check for common characteristic mass fragments. The final manual reference alignment contained 436 peaks grouped into 68 distinct rows.

The overlap of the three reference multiple alignments is visualized in Figure 4.17. As expected, the MGMA reference is a perfect subset of the GMA reference. More interestingly, there is a large overlap of the manual reference with the automated methods' reference alignments, supporting the claim that these automated methods

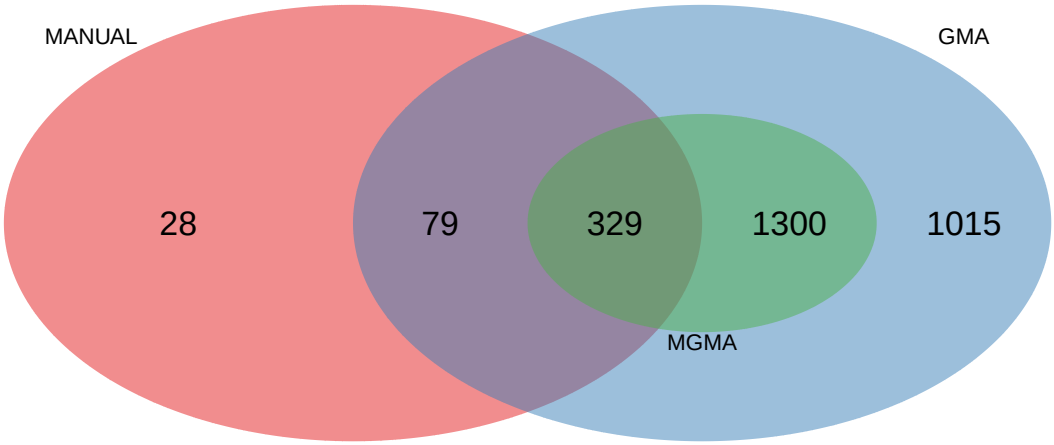


Figure 4.17.: Euler diagram of the peak set overlap for CHLAMY dataset I for GMA, MGMA, and manual multiple alignment reference generation.

for reference generation based on the assigned compound names capture most of the peak alignments contained in the manual reference alignment. A small proportion of 28 peaks (6.4% of peaks in the manual reference) occur exclusively within the manual alignment and not within any of the automated methods. Of these, 24 peaks were differently assigned in the automated methods versus the manual reference, whereas 4 peaks were not reported at all by those methods. 149 peaks were found missing from the manual reference in comparison to the GMA reference generation method, due to the stricter selection criteria that were employed in order to exclude potential false positive peak assignments.

The manual reference alignment, the peak reports for each sample as exported from ChromaTOF, and the raw data files in netCDF format, are available as dataset MTBLS37[6] from the MetaboLights database (Haug et al. 2013).

Results for CHLAMY Dataset I

All three BIPACE variants using either the cosine or Pearson's linear correlation as similarity functions between mass spectra perform better than any GUINEU, MSPA, or SWPA variant (see Figure 4.18). BIPACE 2D achieves F1 scores of 0.6692 (GMA reference), 0.7662 (MGMA reference), and 0.7429 (MANUAL reference), with $D_1 = 100$ s, $D_2 = 0.5$ s, thresholds $T_1 = 0.99$ and $T_2 = 0.99$, together with

6. http://www.ebi.ac.uk/metabolights/MTBLS37

an *MCS* value of 2. The F1 values are visualized in Figure 4.18. BiPACE 2D also achieves the highest Recall values of 0.5752 (GMA), 0.7079 (MGMA), and 0.7596 (MANUAL) while still maintaining reasonable values for Precision between 0.72 and 0.835 (see Appendix C.5).

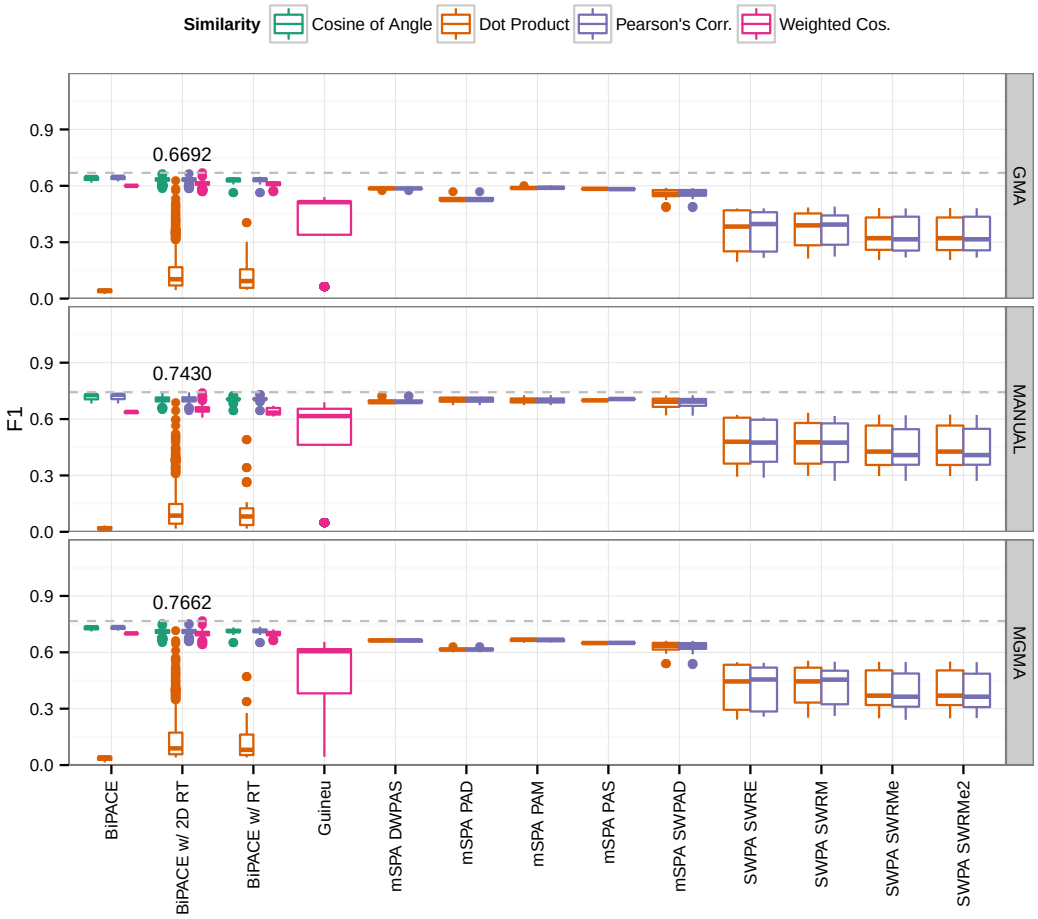


Figure 4.18.: F1 score for all parameterizations of the evaluated algorithms for CHLAMY dataset I. BiPACE 2D performs clearly better than any of the other methods using the dot product as pairwise similarity between mass spectra.

The considerably low values for Recall achieved by the different methods may be due to the complexity of the biological samples and the large number of very closely related peaks and associated peak areas. The best average pairwise $F1_p$ scores are also achieved by BiPACE 2D (GMA: 0.7198 ± 0.0335, MGMA: 0.8293 ± 0.0247, MANUAL: 0.864 ± 0.0606), with GUINEU (GMA: 0.6434 ± 0.0419, MGMA: 0.7766 ± 0.0338, MANUAL: 0.8481 ± 0.0528) placing second and different MSPA variants

placing third (GMA, mSPA-PAM: 0.5976 ± 0.1283, MGMA, mSPA-DWPAS: 0.6907 ± 0.1296, MANUAL, mSPA-SW-PAD: 0.7475 ± 0.1741).

Table C.4 on page 233 holds more detailed results for CHLAMY dataset I. Individual parametrizations of the best instances are again available online[7].

4.5. Conclusions

We have shown that BiPACE 2D is a competitive algorithm that is able to achieve better precision and recall, as well as F1 and average pairwise $F1_p$ score values in comparison to mSPA, SWPA, and GUINEU on three out of the four evaluated datasets. These three datasets were all acquired under homogeneous conditions, while the dataset where BiPACE 2D did not outperform mSPA and GUINEU was acquired under heterogeneous conditions. Thus, BiPACE 2D should ideally be applied to data acquired under the same conditions, but due to its low false positive rate, it may still be a valid alternative for data acquired under heterogeneous conditions as well.Concerning the parameters for BiPACE 2D, the weighted cosine appears to be the most sensitive mass spectral similarity and should therefor be used as the default. The *MCS* parameter was set to the minimum size of 2 in all evaluated parameterizations, thus leading to all cliques being reported by BiPACE and its variants. The $D1$ and $D2$ parameters should be set according to the expected retention time standard deviation of the samples under comparison, in separation dimensions one and two, respectively. Finally, the threshold parameters $T1$ and $T2$ allow for fine-tuning of the sensitivity of the algorithm, where higher values exclude potential matches earlier during the pairwise similarity calculation phase of BiPACE. It is further notable that BiPACE 2D was on average 3 to 10 times faster than any of the mSPA or SWPA variants for the larger and more complex datasets (SWPA dataset I and CHLAMY dataset I, see Appendix C for details), while consuming less memory. GUINEU achieved comparable speed, but required more memory. We have demonstrated the applicability of BiPACE 2D to small datasets with a few compounds, as well as to larger datasets with hundreds to suspected thousands of different compounds. BiPACE's pairwise similarity calculation can be run in parallel using multiple CPU cores to speed up its runtime. It has been successfully tested on 250 files containing 100,000 peaks on commodity hardware within 9 GBytes of random access memory. This qualifies BiPACE 2D as a good candidate for automated medium to high-throughput applications in the field of metabolomics and analytical chemistry.

We have further introduced a fast yet sensitive method for peak location and filtering that can be used to generate input peak lists for BiPACE 2D. However, this method still lacks a thorough evaluation on diverse GC×GC-MS datasets and a comparison to the results achieved with other peak finding methods, such as the method available in the propriertary vendor software ChromaTOF (LECO Corp., St. Joseph, MI, USA).

7. http://maltcms.sf.net/pub/bipace2d

5

Maltcms

We have already introduced methods for one- and two-dimensional chromatography-mass spectrometry data in the previous chapters and have compared features between specialized pipelines of the **M**odular **A**pplication **T**oolkit for **C**hromatography-**M**ass Spectrometry (MALTCMS) and other Open Source software. In this chapter, we describe the architecture and additional features of MALTCMS that have not been mentioned yet.

We begin with a short summary of the requirements for MALTCMS and its underlying framework, the **C**ommon **R**untime **O**bject **S**upport **S**ystem (CROSS) in Section 5.1. We then describe CROSS and the parallelization framework **M**altcms **Pa**rallel E**x**ecution **S**ystem (MPAXS) in detail and explain some of CROSS' central data structures.

We close this chapter with an overview of the domain-specific implementation MALTCMS, that is based on CROSS, in Section 5.2. We describe the main data structures implemented in MALTCMS and explain the different parts of functionality that it provides.

5.1. Cross

CROSS[1] was originally designed and implemented during work on CHROMA (Hoffmann and Stoye 2009) as the basis for a server-side program for alignment and visualization of GC-MS data with dynamic time warping (DTW). It is designed for the definition, creation and execution of sequential workflows of *fragment commands* that realize partial functionalities in a typical data processing workflow, such as the one defined at the end of Chapter 2.

During the design and implementation of CROSS, we identified the following requirements:

- Platform independence;

1. http://sf.net/p/maltcmscross

- Possibility for headless operation on servers without a graphical frontend;
- Simple and accessible application programming interface (API);
- No domain-dependent implementations (realized by MALTCMS);
- Modularity through custom implementation of API interfaces and abstract classes;
- No explicit requirement for a database environment;
- Dynamic implementation discovery through JAVA's *ServiceLoader* facilities;
- Explicitly linear workflows for easier validation and accessibility;
- Self-describing workflow output for documentation and archivability.

These requirements allow CROSS to be embedded in either server-side or client-side programs as a library, allowing for easy customization and extension.

CROSS has been implemented in the JAVA (Gosling 2013) programming language, version 7. JAVA is an object-oriented programming language that is compiled into intermediate bytecode. The bytecode is then executed by a virtual machine that is operating system specific. The virtual machine interprets the JAVA bytecode and accelerates heavily used parts of it by compiling the respective bytecode just-in-time (JIT compilation) into platform-specific native code. Through optimization of the bytecode, JAVA is nowadays often on par in terms of execution speed when compared to C (Kernighan and Ritchie 1988) or C++ (Stroustrup 2013). However, its object-based type system introduces additional overhead that generally leads to higher memory requirements than C++.

JAVA was designed to combine portability, networking features, security and concurrency with a simpler and more robust programming interface than C or C++. It therefore does not allow direct access and manipulation of the program's memory via pointers, as C and C++ do, decreasing the risk for unintentional security flaws by memory access violations.

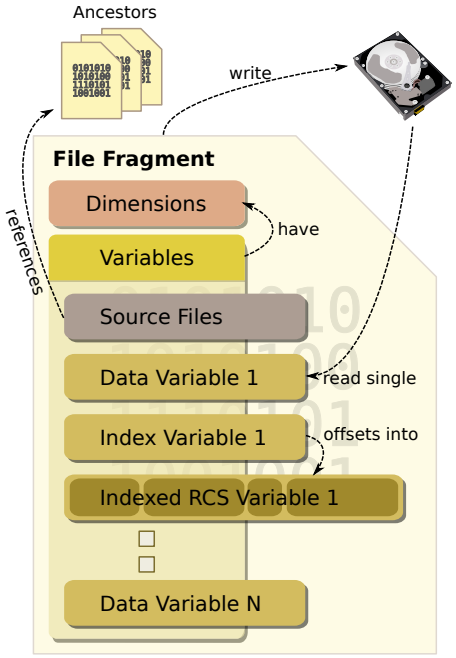
Furthermore, JAVA already includes a very diverse standard library of objects and methods for networking, input-output operations, parallelization, remote method invocation (RMI), modularization (ServiceLoader), and concurrent and non-concurrent data structures to exploit modern computers with multiple central processing units (CPUs) and multiple CPU-core architectures.

5.1.1. Core Data Structures

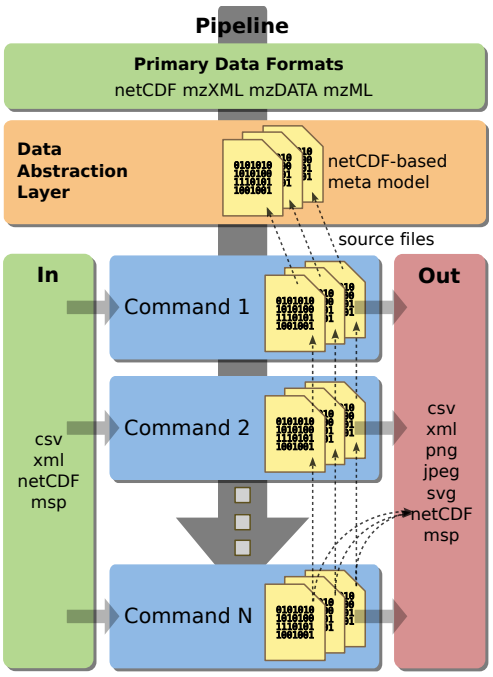
The most important data structures in CROSS are the *FileFragment* and its relative, the *VariableFragment*. Both represent parts of data files that are either accessed from disk or a network resource (read-only mode) or that are cached in memory or on disk during creation (read-write mode).

A file fragment is an aggregation of variable fragment objects, identified by a uniform resource identifier (URI)[2], pointing to either a local or remote storage location,

2. http://tools.ietf.org/html/rfc3986



(a) A file fragment contains dimensions that name common extents of the variables contained in the file fragment. Variables are named and contain the actual data in the form of one- or multidimensional arrays. Each variable may reference any number of dimensions, stating a named range of each of its data array's dimensions that can be shared with variables that have the same range and are thus semantically related. File fragments are created in memory with optional disk-based caching of individual variable data before they are written to disk. When saved file fragments are accessed, only their structure is initially loaded into memory. Array data is only loaded upon explicit request from a variable and may be cached for faster access. A file fragment can reference an arbitrary number of other file fragments via the source files variable. Variables may serve as an offset index into other variable data arrays that have been saved in RCS format as a dense representation of sparse data. Indexed variable data can then be retrieved as a collection of arrays for access to each of the records stored in RCS format.



(b) The main data flow is oriented from top to bottom, based on the individual file fragments (source files) as input to the workflow. Auxiliary data can be passed into the workflow for each fragment command individually (In). Created data are stored in a workflow-specific location, below a fragment command-specific directory (Out), and are recorded in the workflow's log. Each command operates on the input files passed to it from its predecessor, or from the pipeline's input in case of the first command. Transitions between commands serve as checkpoints for the workflow and are used to update the workflow log.

Figure 5.1.: Cross File Fragment and Pipeline. a) Schematic of the file fragment data structure. b) Schematic of a Cross-based workflow backed by a linear processing pipeline.

such as a public web server. File fragment objects may reference an arbitrary number of source files, thereby allowing virtual aggregation of processing result variables of previous fragment commands. Shadowing allows file fragments to hide the existence of an upstream variable of the same name from downstream file fragments (see Figure 5.1(a)). Data source implementations allow different URI extensions to be handled, so that file fragment objects can exist as simple files on disk or within a distributed database system.

Cross provides a thin layer on top of the netCDF (Rew and Davis 1990) JAVA library[3] in order to define, populate, and modify a file in memory with intermediate caching, before it is written to file in one piece. Additionally, access to variable data is handled agnostic to the actual file format backing the physical file by the data source implementations that map the respective variable names to the ones suitable for their supported data format. The loading of data from file is performed lazily, with the option of caching the array data for faster repeated access. Variables also support the chunked retrieval of their data to enable access to arrays in size that exceed available main memory. This is primarily used for the access to GC×GC-MS raw data files that can easily exceed four GB in size.

5.1.2. Mathematical Utilities and Statistics

The mathematical and statistical methods available in Cross and Maltcms are provided by the Apache Commons Math library[4]. This library provides the t-test (one- and two-sided, paired and unpaired) for comparisons between two groups, and one-way ANOVA for the comparison of multiple groups. Additionally, it provides access to common distributions, and descriptive statistics like quantiles, mean and median of empirical distributions. The LOESS method used in Section 5.2.3 for baseline estimation is also provided, among other interpolation methods, by the same library.

Efficient methods for linear algebra and appropriate dense and sparse data structures are provided by the Colt library[5].

5.1.3. Workflow Model

The *CommandPipeline* class is the main implementation that holds the *AFragmentCommand* instances for execution. It is invoked, processed, and monitored by the *DefaultWorkflow* implementation. Cross provides an abstraction of a linear workflow, limited to the transformation or rearrangement of input data into output data (see Figure 5.1(b)). The relationship of input to output data in Cross can be determined individually by each fragment command. A typical command would use a one-to-one relationship between in and output file fragments, for example introducing a new variable, thereby augmenting the corresponding input fragment. The number

3. http://www.unidata.ucar.edu/software/thredds/current/netcdf-java
4. http://commons.apache.org/proper/commons-math
5. http://acs.lbl.gov/software/colt

of in and output file fragments processed by a fragment command can differ, thus allowing map-reduce-like processing schemes or generally schemes with different or equal parities, for example producing one result file from multiple input files. Individual workflows can be connected by running a different configuration on the output of a previously executed workflow. This also allows to create very large, implicit processing networks, for more advanced use-cases.

A workflow in Cross is made up of a sequence of fragment command objects that use file fragments as their in- and output type (see Figure 5.1(b)). Each fragment command can itself represent a command sequence of other fragment commands, essentially executing a sub-workflow within its output directory. Fragment commands can run parallel or sequential computations by implementing *java.util.Callable<T>*, a typed unit of computation that returns objects of type *T* upon completion. The parallel execution can be delegated to the Mpaxs framework for distributed computation in a grid system (see Section 5.1.8).

The end of execution of a fragment command serves as a checkpoint to the workflow monitoring the executing pipeline. Before a fragment command returns control to the workflow, all pending jobs or sub-pipelines started by it must have terminated. The workflow log is continuously updated during execution of a fragment command, noting created resources and monitoring overall progress. Failure of individual commands leads to an early and safe termination of the workflow.

The basic configuration of all workflow elements is performed using a Spring application context created from an XML configuration (see Section 5.1.7), supplemented by runtime properties.

Validation

Each fragment command can state its required variables explicitly using the class-level annotations RequiresVariables, RequiresOptionalVariables, and it can state the variables it provides to downstream commands using the annotation ProvidesVariables. When a workflow instance is created from its configuration, the workflow can optionally be checked for all required variables to be either provided by input data fragments, or by at least one of the fragment commands. The checking is performed in the order of declaration of the fragment commands. Thus, a fragment command requesting a specific variable must be preceded either by other fragment commands providing the required variables, or by input data that contains those variables directly or transitively via its ancestor files, as defined in each fragment file's source_files variable.

Monitoring and Transformation

A workflow monitors the fragment commands it executes and notifies registered listeners of various workflow-related events. These include the creation of primary and secondary processing results, as well as general progress information. A workflow

logs all completed tasks and their results in a distinct and unique (depending on configuration) self-contained (except for initial input data) output directory. This output directory contains all information necessary to re-run the workflow with the exact same parameters and conditions. Workflows in CROSS are therefore self-descriptive and repeatable.

5.1.4. Caching

CROSS provides a number of different caches for array data and higher-level data structures for objects that are often required or expensive to create. Currently, Ehcache[6] is used to provide volatile in-memory and non-volatile, disk-based caches. Additionally, CROSS has a volatile in-memory cache based on *SoftReferences* for access to array data. *SoftReferences* are released automatically by the JAVA virtual machine when available heap memory runs low and thus allow the implementation of a cache that exploits available memory as much as possible.

5.1.5. Modularity

CROSS locates available implementations for API classes using JAVA's *ServiceLoader*[7]. This facility allows discovery of implementations of interfaces at runtime and is used by CROSS for the discovery of fragment command implementations, data source implementations, and controlled vocabulary providers.

CROSS additionally exposes OSGi-compliant information for deployment in modular systems based on OSGi[8], such as the integrated development environment Eclipse[9] or JAVA application servers like GlassFish[10]. OSGi support was added to CROSS, MALTCMS, and some of their dependencies with the help of Tobias Placht.

For direct deployment in NetBeans Rich Client Platform applications[11] like MAUI (see Chapter 6), the CROSS and MALTCMS modules are also bundled in NetBeans module format (nbm).

5.1.6. Controlled Vocabulary

CROSS variables have simple string-based names. However, in different contexts, the same variable name can have a different meaning. Thus, CROSS supports namespaced, controlled vocabularies (CVs) for specific domains that translate a variable placeholder name with a namespace (here: andims), such as andims.var.total_intensity to the actual, CV-resolved name: total_intensity. The resolution mechanism can be used for dimension names (andims.dimension.), attribute names (andims.attr.), unit names (andims.units.), and variable names (andims.var.). The default namespace used

6. http://www.ehcache.org
7. http://docs.oracle.com/javase/7/docs/api/java/util/ServiceLoader.html
8. http://www.osgi.org
9. http://www.eclipse.org
10. https://glassfish.java.net
11. https://netbeans.org/features/platform/

by Cross and Maltcms is empty, e.g. var.source_files is resolved to source_files. CV providers can hook into Cross via the *ServiceLoader* functionality and reserve their own namespace for resolution. The CV support in Cross also allows deprecation of variable names that have been replaced by a more concise term to alert users and developers of possible incompatibilities.

5.1.7. Inversion of Control Container

Cross uses the Spring[12] framework as a configurable factory for the assembly of object graphs. Spring uses the inversion of control (IoC) paradigm to create, connect, and configure other objects without their explicit knowledge of the factory. Instead, each object can rely on its requirements to be fulfilled at runtime by the factory, thus control is inverted, away from the object, and managed by the application context factory. This simplifies the use of such a container, as most objects can be rather simply structured plain old JAVA objects (POJOs). The assembly of the actual application is performed by the IoC framework, using the dependency description within either a textual XML file or as defined by annotations on the objects. The framework may even be able to automatically assemble the object graph of an application if required dependencies can be uniquely fulfilled by another object. This process is called *auto-wiring*.

In case of Cross, Spring is mainly used for the construction of the processing workflow and pipeline objects along with supporting objects and the fragment commands. The latter ones perform the actual data processing, while the workflow and pipeline objects provide the required infrastructure for validation, execution, and auditing of intermediate results.

5.1.8. Parallelization

Cross uses the Mpaxs framework[13] for transparent parallelization of *Runnable* and *Callable* tasks either within the local virtual machine or on other remote machines that are coordinated through remote method invocation (RMI), a JAVA-specific variant of remote procedure calling (RPC). Mpaxs was originally developed during the Bachelor's thesis of Kai Bernd Stadermann for grid-based parallel processing. A computing grid is a heterogenuous network of computers that are controlled by a scheduling system. Users that want to execute programs on a grid computer submit their job to the batch submission system. The scheduling system then assigns the job to run on an available grid computer, depending on its workload and other parameters. In case of Mpaxs, such a grid system should be compatible to the standards defined by the Open Grid Forum[14] and provide an implementation of the distributed resource management application api (DRMAA) for JAVA.

12. http://projects.spring.io/spring-framework
13. http://sf.net/p/mpaxs
14. http://www.ogf.org

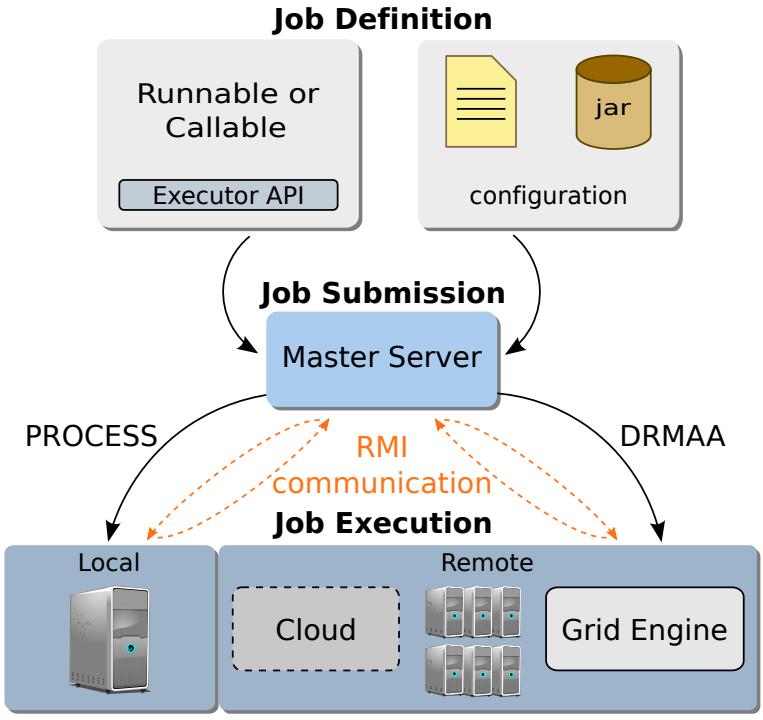


Figure 5.2.: Schematic of parallel processing with MPAXS.

Architecture

The basic components of MPAXS are the *master server*, that runs within the same process as the user's software, and an arbitrary number of *compute hosts* that run either locally as a separate process on the same computer or remotely on physically different hosts within a local network (see Figure 5.2). The master server and the compute hosts form a dynamic ad-hoc network, if an infrastructure for launching of compute hosts is available, such as a DRMAA-compatible grid engine implementation like Open Grid Engine[15], Torque[16] or others. Otherwise, the compute hosts have to be started manually by the user or local administrators. Communication and transfer of data between the master server and the compute hosts is handled via RMI and serialization. *Serialization* is the process of transforming an object to a binary representation, while *deserialization* is the opposite process to obtain an object from its binary representation. All classes that should be amenable for parallelization have to be either *serializable*, using JAVA's default serialization protocol, or be *externalizable*, implementing their own custom serialization protocol.

In the current implementation, the communication between server and compute hosts is not encrypted since MPAXS is primarily operated within a local grid. The compute host however uses a unique session token for identification with the clients.

15. http://gridscheduler.sourceforge.net
16. http://www.adaptivecomputing.com/products/open-source/torque

This token is required for every remote action performed, either from server or client side, to avoid accidental hijacking of compute hosts by other master server instances in the same grid environment.

Compute hosts are created on demand by the master server, up to a pre-configured maximum number. The hosts are configured with an initial slot capacity, corresponding to the maximum number of concurrent jobs that should be runnable on each host. Then, jobs that are submitted for execution to the master server are distributed to each available compute host using a fair round robin scheduling algorithm. After a job has terminated, either normally or abnormally, the compute host reports the job's status back to the master server and also registers the job's slot as available. Compute hosts terminate and shut down after a user-defined time out period has passed if they do not receive jobs. The compute hosts query the master server in defined time intervals to see, whether he is still alive. They terminate after a given timeout if they do not receive a reply from the master server to avoid orphan compute hosts in the system.

The code that the compute hosts should execute does not have to be immediately available, but rather should be made available below a given URL that needs to be passed to the compute hosts on start up (the code base location). This URL can point to a local or remote directory, e.g. on a public web server, that contains the required libraries as jar files (zipped JAVA bytecode).

MPAXS abstracts computational units as *jobs*. The master server scheduler submits jobs to the next compute host registered with a positive workload capacity. Jobs have a default priority that can be increased to allow scheduling of the job prior to jobs with the next lower priority. They are initially submitted to an unbounded, concurrent priority queue by the user and inserted according to their priority. Jobs that should run repeatedly in defined intervals can be submitted as scheduled jobs.

Integration with JAVA's Concurrency Utilities

In JAVA, a concurrent action can be implemented in one of two ways. Traditionally, *Runnable* was the base interface for concurrent implementations with a single method *run*. Due to the lack of a return type of *run* it complicated the implementation of side-effect free concurrent programs. Thus, in JAVA version 5, the *Callable*, with a generic return type of its *call* method was introduced for custom concurrent implementations.

Each job in MPAXS needs at least a *Runnable* or *Callable* implementation, and optionally, a configuration file, if the job is supplied as a jar archive. The MPAXS-spi module provides implementations of JAVA's parallel *ExecutorService* for simplified integration. Additionally, the *CompletionService* and *ResubmissionCompletionService* offer the additional functionality of monitoring each submitted job's status and possible exceptions and failures during execution within the grid environment. They both wrap JAVA's *ExecutorCompletionService* but appear as standard *Callable* implementations that allows to submit them to local, non-distributed *ExecutorService* implementations. The *ResubmissionCompletionService* additionally allows to set a maximum resubmission limit for jobs that fail due to random errors or timeouts

within the grid system. Both implementations allow for a blocking operation that waits for all jobs to complete or terminate abnormally, before returning their results and a list of the failed jobs for further inspection.

Alternatively, upon submission to the master server, each job immediately returns a custom and RMI-aware *Future* object that represents the result of the future computation to be executed by MPAXS. This allows users of the API maximum flexibility when designing parallel algorithms that should monitor each job's status individually and react to results as soon as they become available.

5.2. Maltcms

MALTCMS[17] is a domain-specific implementation for chromatography-mass spectrometry and related fields, based on CROSS.

Figure 5.3 shows the layers and subsystems upon which MALTCMS is based. The third party layer provides common and basic functionality, e.g. for charting, mathematics, caching and distributed computing. CROSS, as already introduced, provides a number of modules for caching, pipeline and workflow functionality, file fragment and basic data structure implementations, as well as the data source (IO provider) framework. MALTCMS provides high-level data structures for chromatograms, mass spectra, metabolites, alignment anchors, peaks and peak groups. It furthermore provides different commands and individual modules for chromatogram and mass spectra preprocessing, peak detection and integration, peak and chromatogram alignment, visualizations, statistical tests, and putative peak identification. We have already described and compared the features of MALTCMS that specifically apply for GC-MS in Section 3.1 and GC×GC-MS and Section 4.1. We therefore focus our description here on the features that have only been mentioned briefly before.

5.2.1. Data Structures

The main domain-specific data structures provided by MALTCMS are provided in the *maltcms-datastructures* module. Data structures for *chromatograms* abstract the low-level structure of the file fragment-based data model as defined by CROSS and allow easier access to scan objects that model mass spectra with additional retention time information. Scans are implementations of the more general concept of *feature vectors* introduced in MALTCMS to provide both specific and efficient direct access to resources provided by the feature vectors, as well as to provide generic access to resources based on defined names, similar to the variable names used in file fragments. The feature vectors are serializable and thus amenable to parallelization and disk-based caching. One- and two-dimensional chromatograms are created from the corresponding file fragment and provide iterators for efficient, cached access to mass spectra and other information, abstracted as one- or two-dimensional scan objects, respectively. Random access to scans is also possible and eventually cached.

17. http://maltcms.sf.net

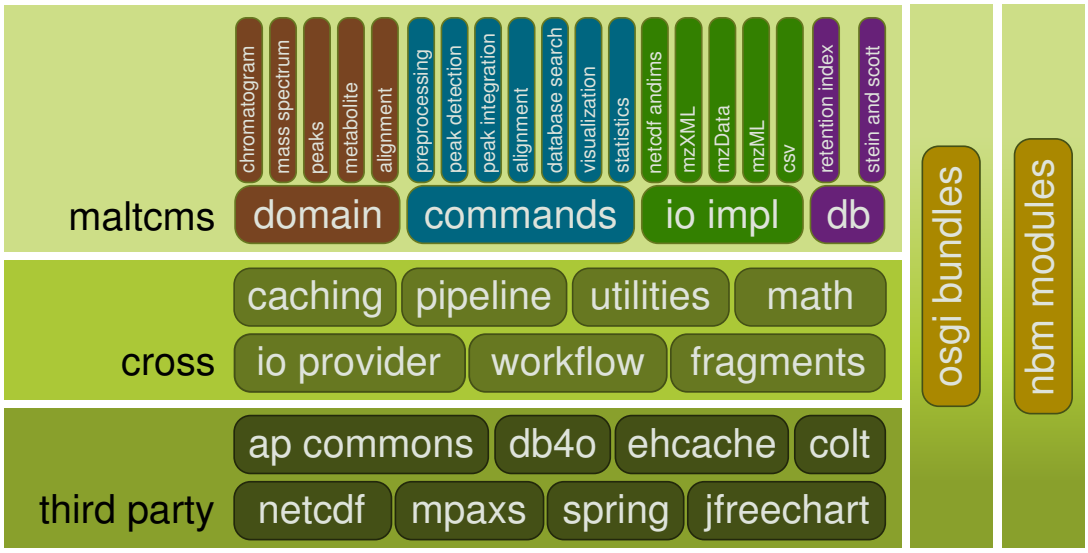


Figure 5.3.: Software layers and subsystems of CROSS and MALTCMS. CROSS provides interfaces and some default implementations for the domain-specific functionality realized by MALTCMS.

Especially for the large GC×GC-MS and comprehensive two-dimensional liquid chromatography-mass spectrometry (LC×LC-MS) data sets, scan data is loaded in larger chunks that are aligned to the modulation period to allow for efficient anticipatory access to neighboring scans. Chromatograms also provide methods to iterate over only a selected time range and to retrieve a subset of scans for a specific MS fragmentation level.

Peaks are modeled as specialized feature vectors, storing information about their area and baseline, as well as their retention time, signal-to-noise factor, area normalization methods used to calculate their normalized area, and their type. The peak type indicates whether a peak was integrated from the raw or filtered TIC or EIC signals. A *peak group* can also contain other peaks. Since the peaks in the peak group can also store a portion of the integrated signal, either TIC or EIC, peak groups can be used to obtain a reconstructed group mass spectrum and other properties of the peak group, like its mean retention time. Thus, they can hold the result of deconvolution for individual ion channels over the full integration range of the peak group.

Other data structures available in MALTCMS provide support for alignment maps, sparse and dense arrays that are used for the CEMAPP-DTW calculations, as well as data structures for metadata handling, like experiments and sample groups.

5.2.2. Filtering and Normalization

The data processing tasks in MALTCMS often involve the processing of array data. Especially for peak finding, a number of filters are available, providing for example the CWT peak finder (see Section 4.2.1). Other available filters include the Savitzky-

Golay filter, moving average and median filters, as well as morphological filters like the top-hat filter (Lange et al. 2007). These filters are available independently of the other data structures used in MALTCMS.

5.2.3. Peak Detection and Integration

MALTCMS contains a method for location and integration of peaks in one-dimensional TIC data. The same method can repeatedly be applied to locate and integrate peaks from multiple EICs as well. However, proprietary vendor software usually has more context data available from the machine that was used for sample acquisition, as well as having been optimized for a long time, which is why we rely mainly on vendor provided peak lists in MALTCMS. The TIC-based peak finding that we describe here, was primarily used on GC-FID data (see Section A.3 for more details).

The peak finder employs a configurable sequence of initial filtering of the signal. By default, it uses a Savitzky-Golay filter (Savitzky and Golay 1964) for signal smoothing. It then locates minima within the signal in order to fit a non-linear baseline using the LOESS method. This baseline is then used to estimate the local signal-to-noise ratio for each point of the signal. Peaks are then detected by searching for local maxima that exceed the user-defined signal-to-noise threshold and that are separated by a user-defined minimum number of scans. Peak integration is performed either on the baseline-corrected or on the raw signal. Peak bounds are determined by inspecting first, second and third-order derivatives of the filtered signal. Peaks and areas are reported in ANDI-CHROM peak format, in MALTCMS feature XML format, and in a simple CSV format. Figure 5.4 shows the result of peak finding applied to GC-FID data, as visualized by MAUI (see Chapter 6).

Peak Integration for GC×GC-MS Data

Based on the peak seeds that the CWT-based peak finder reports (see Section 4.2), or based on peak locations provided by other methods, the ChromA4D pipeline available in MALTCMS uses a variant of seeded region growing (SRG) (Adams and Bischof 1994) to locate two-dimensional peak bounds. This method addresses some of the problems in GC×GC that are encountered with the related watershed segmentation algorithm (Vivó-Truyols and Janssen 2010; Latha, Reichenbach, and Tao 2011) by including mass spectral information during extension of the peak bounds from the initial seeds. We are interested in finding the area of the reported peaks for downstream quantitative comparison of samples within and between experimental conditions. Since we also have the rich mass spectral information available, we use that information in order to find the bounds of each peak by comparing the mass spectra of neighboring feature vectors to the peak apex mass spectrum, based on a suitable similarity. Mass spectrum similarities that are suitable for this task have already been introduced in Section 3.3. Very similar neighboring peaks may in fact originate from one chromatographic peak, but have been detected as separate peaks in different modulation periods. The enhanced SRG method fuses

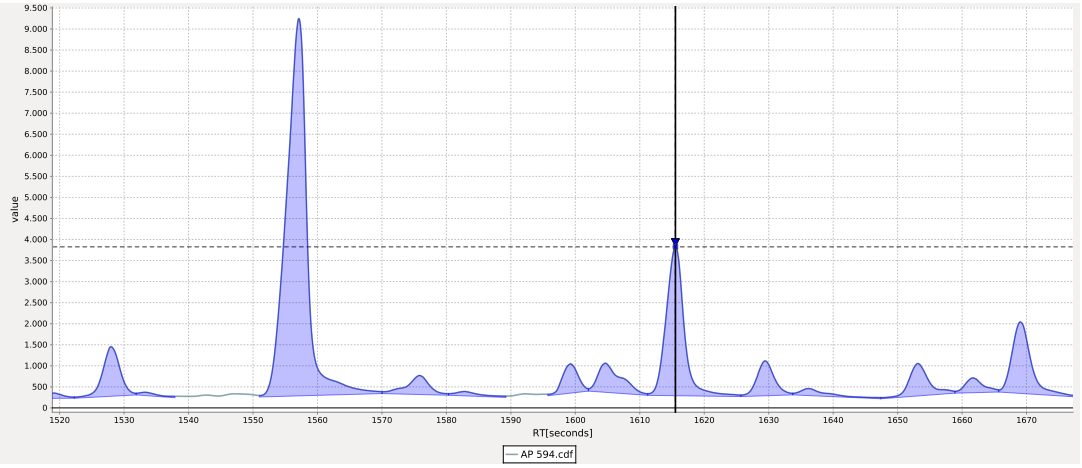


Figure 5.4.: Result of TIC Peak Finder on GC-FID data using a Savitzky-Golay filter with window width of 25 (2×12 + 1) points, LOESS baseline estimation using local sliding window samples over 1000 points and a bandwidth of 0.3 with two robustness iterations. Local SNR calculation used a threshold of 3, and a minimum apex-to-apex peak separation window of 20 scans was used. Peak integration was performed over the raw, uncorrected TIC (visualization shows only the estimated baseline).

the corresponding peak areas in both retention time dimensions if the similarity between the representative mass spectra of each peak area is above a user-defined minimum similarity threshold. Figure A.4(a) on page 181 shows the result of the SRG and region merging on a GC×GC-MS dataset. The peak integration can be performed after the bounds have been determined and neighboring peak areas above the threshold have been fused. It is possible to integrate either the full intensities of each mass spectrum within the peak area, or to integrate only the intensities of a representative or significant subset of the masses occurring within the peak bounds, possibly excluding masses related to TMS derivatization. We explicitly exclude baseline and noise estimation in this context, since they can easily be realized as preprocessing steps that lead to a baseline and noise-corrected GC×GC-MS dataset. The seeded region growing and merging methods were implemented by Mathias Wilhelm.

5.2.4. Alignment

We already described most of the alignment algorithms available in Maltcms, like BiPACE, CeMAPP-DTW, and BiPACE 2D in Sections 3.3, 3.4, and 4.3. An additional method based on DTW for two-dimensional TIC images from GC×GC-MS data was developed and implemented by Mathias Wilhelm during the work on his Bachelor thesis. Figure A.4(b) shows the result of aligning the 2D TICs of two GC×GC-MS chromatograms using this variant of DTW as a pseudo-color differential plot.

5.2.5. Visualization

In order to visualize the various data structures and algorithm results, MALTCMS provides convenient methods to create false color images of similarity matrices with custom color palettes and of 2D-TIC surfaces from GC×GC-MS data (see Figure A.3(a) on page 180). Additionally, based on the JFreeChart library[18], overlay and co-plot visualizations of unaligned and aligned chromatogram TICs and EICs are available (Hoffmann and Stoye 2009). These plots are complemented by statistical box-and-whisker charts that are used for the visualization of retention time deviations of aligned peak groups in BIPACE. Charts can be saved to common bitmap formats like JPEG or PNG. Since MALTCMS was designed to run without user-interaction, all plots and charts are rendered in headless, offscreen mode.

5.2.6. IO Provider Implementations

The CROSS IO provider framework requires that a file format can be represented as a collection of named variables with additional content. Furthermore, variables may define dimensions, which can be shared among variables to indicate similar ranges or coordinate systems, following the recommendations of Unidata for a defined, common data model (CDM) in netCDF files[19]. For MALTCMS, the minimum required variables are defined by the ASTM ANDI-MS and ANDI-CHROM standards (Erickson 2000). Table 5.1 gives an overview of these variables and their data types.

netCDF

All variables contained in a netCDF file can be directly accessed by MALTCMS. For the datasets following the ANDI-MS conventions, however, each vendor exports the data in slightly different ways, sometimes with uninitialized variables that contain default values. Thus, MALTCMS requires only a minimal subset of the variables defined in the ANDI-MS standard for operation. For GC×GC-MS, no standard was available when support for it was added to MALTCMS (see Table 5.2). Recently, however, the standardization of mzML and its controlled vocabulary has been amended for metabolomics and analytical technologies that play an important role for it (see Table 5.3). We have therefore adapted the terms used in MALTCMS to those used in mzML. A complete list of supported variables based on the ANDI-MS and ANDI-CHROM standards is located in the cfg/cv directory within the MALTCMS distribution[20].

18. http://www.jfree.org/jfreechart
19. http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/CDM
20. available at sf.net/p/maltcms/files/maltcms

Table 5.1.: Overview of the ANDI-MS variable subset used by MALTCMS. Optional variables are marked with (*). Variables with data type *float* can be stored in either single or double precision. a, b: var.scan_index contains scan offsets for RCS storage into var.mass_values and var.intensity_values. c: ms_level is originally part of the mzML CV, but has been included in the MALTCMS data model for preliminary access to MS/MS data.

MALTCMS CV Name	Type	Data Type	Dimension
var.total_intensity	detector count	integer	scan_number
var.scan_acquisition_time	time	float	scan_number
var.scan_index	scan offsets into [a,b]	integer	scan_number
var.mass_values[a]	mass	float	point_number
var.intensity_values [b]	detector count	integer	point_number
var.mass_range_min*	minimum mass of ms	float	scan_number
var.mass_range_max*	maximum mass of ms	float	scan_number
var.ms_level*,c	ms fragmentation level	integer	scan_number

Table 5.2.: Overview of the variable subset used by MALTCMS for two-dimensional chromatography. Variables with data type *float* can be stored in either single or double precision.

MALTCMS CV Name	Type	Data Type	Dimension
var.modulation_time	modulation time	float	modulation_time
var.scan_rate	ms acquisition rate (Hz)	float	scan_rate
var.first_column_elution_time	1st column time	float	scan_number
var.second_column_elution_time	2nd column time	float	scan_number
var.total_intensity_2d	2D TIC	float	modulation_time * scan_rate

mzXML

The mzXML format (Pedrioli et al. 2004) is supported as a read-only data source via the JRAP library[21] that is part of the Sashimi project under the stewardship of the Seattle Proteome Center (SPC). Support for this file format is considered deprecated and we currently advise users to use the msconvert program from the ProteoWizard project (Kessner et al. 2008) to convert mzXML files to mzML format.

mzData

The mzData format (Orchard et al. 2005) is supported by a custom binding using the JAVA architecture for XML binding (JAXB)[22]. It currently provides read-only access to mzData files. Support for this file format is also considered deprecated and we currently advise users to use the msconvert program from the ProteoWizard project (Kessner et al. 2008) to convert mzXML files to mzML format.

mzML

Maltcms supports reading and writing files in the mzML format via the jmzML library (Côté, Reisinger, and Martens 2010). Together with the netCDF-based native file format, mzML is the main supported format for data supplied to and written by Maltcms. In order to standardize the controlled vocabulary (CV) of mzML for metabolomics applications, a number of analytical techniques and terms were recently added to the ontology (see Table 5.3). These include terms for multidimensional chromatography-mass spectrometry and also a term for Maltcms to identify it as the creator of mzML files (available as of Maltcms version 1.3.1). Since writing to mzML involves some non-trivial changes between the data structures used in Maltcms and jmzML, the required functionality is implemented in a custom fragment command (*MZMLExporter*) and not within the data source implementation itself.

OpenMS Feature XML

The OpenMS (Sturm et al. 2008) framework stores features, e.g. picked mass spectral peaks, in an XML format. Maltcms provides a binding to the feature format using JAXB generated classes that allows reading and writing of the format.

Comma and Tab-Separated Data

Comma or tab-separated value (CSV, TSV) data can be accessed and written by the classes CSVReader and CSVWriter. CSV and TSV data can not be mapped directly to the Maltcms data model.

21. http://sashimi.sourceforge.net/jrapdoc
22. https://jaxb.java.net

Table 5.3.: Selection of controlled vocabulary terms in mzML for chromatography-mass spectrometry and mapping to MALTCMS Variables. a: m/z and intensity arrays are binary data arrays. b: These terms are multidimensional chromatography modulation descriptors.

mzML CV Name	ID	Type	Child of	MALTCMS CV Name
ms level	MS:1000511	attribute	spectrum	
binary data array[a]	MS:1000513	array of values	-	-
m/z array	MS:1000514	m/z values[a]	spectrum	var.mass_values
intensity array[a]	MS:1000515	intensity values	spectrum	var.intensity_values
multidimensional chromatography ↪ modulation description[b]	MS:1002084	attribute	run	
GC×GC with fixed modulation time	MS:1002085	value of [b]	-	-
GC×GC with discrete ↪ modulation time steps	MS:1002086	value of [b]	-	-
LC×LC with fixed modulation time	MS:1002087	value of [b]	-	-
LC×LC with discrete ↪ modulation time steps	MS:1002088	value of [b]	-	-
modulation time	MS:1002042	time interval	run	var.modulation_time
first column elution time	MS:1002082	time	scan	var.first_column_elution_time
second column elution time	MS:1002083	time	scan	var.second_column_elution_time
scan rate	MS:1000015	$time^{-1}$	scan	var.scan_rate
Maltcms	MS:1002344	name	softwareList	application.name
Version	-	version	softwareType	application.version

ChromaTOF Reports

The data provider for LECO ChromaTOF reports does not directly map to the MALTCMS data model. Instead, the data provider can be used to read ChromaTOF reports individually and process them, before results are stored in the MALTCMS data model.

XLS and XLSX Formats

MALTCMS has an IO provider supporting XLS (old Microsoft Excel format) and XLSX (new, XML-based Microsoft Excel format). However, the data source implementation for a particular format has to be provided by a service provider implementation to be discoverable by the *ServiceLoader* mechanism at runtime. Currently, there is only one implementation available for Agilent ChemStation (Agilent, Santa Clara, CA, USA) peak reports in XLSX format.

Data Conversion

MALTCMS is capable of transcoding from netCDF, mzXML, and mzData to netCDF and mzML formats. However, only the minimal subset (see Table 5.1) is supported for these data sources in the current implementation. The libraries used for accessing mzML, mzData or mzML can always be used directly if access to features that are not covered by the MALTCMS data model is required.

5.2.7. Putative Peak Identification

MALTCMS provides a parser for mass spectral database information in the textual MSP format, that is exportable from the NIST's AMDIS software (Halket et al. 1999). The GMD (Hummel et al. 2007) also provides data in that format for non-commercial, academic use. The databases are imported into files managed by the object database system db4o[23] and are represented there as metabolite objects with mass spectrum, potential retention index and various metadata fields. The databases can be queried by the MALTCMS fragment command *EIMSDBMetaboliteAssignment*. Imported databases can be inspected and searched with the *MetaboliteBrowser* application, realized with the help of Rolf Hilker. Peaks in samples can be putatively identified by using retention index information and any of the available mass spectral similarities, such as the plain or weighted cosine. Additionally, metabolite candidates from the database can be further evaluated to match additional criteria based on a generic query pattern language that supports numerical ranges and fuzzy string matching.

23. http://www.db4o.com

6

Maui

In the previous chapter, we described MALTCMS as a software framework for the processing of data from chromatography-mass spectrometry experiments. We now describe a graphical user interface (GUI) **Ma**ltcms **U**ser **I**nterface (MAUI) that provides a user-friendly access to MALTCMS and that provides additional features for interactive data and result exploration, as well as meta-data organization and statistical evaluation of processed results.

In Section 6.1 we describe the background of MAUI's development and define its requirements. We then give an overview of the project model used by MAUI in order to model experiments, samples, peak data, alignments and statistical results in Section 6.2.

We describe the available methods for data import and export for the interaction with other tools in Section 6.3.

In order to provide a good user-experience, MAUI provides a number of comprehensive interactive visualizations for raw and processed data. We describe these visualizations in Section 6.4.

MAUI extends the database support of MALTCMS for putative identification of peak mass spectra by providing views and actions for the custom creation and curation of user databases that can readily be used either by MALTCMS or MAUI. We describe this support in Section 6.7.

We conclude this chapter with the statistical methods that MAUI currently provides for the comparison of metabolite abundances between different samples in Section 6.6.

6.1. Background

During the development of MALTCMS, there often existed the need to visualize and inspect processing results to evaluate the performance of the different algorithms. Therefore, an early prototype of the MAUI was developed as a very simple application based on the JAVA Swing graphics framework for the purpose of visualization.

However, this soon proved to be far from optimal, especially when new visualizations had to be added. Thus, we evaluated different module systems available for the JAVA platform in order to achieve a better structuring of the user interface components. The two most renowned competing module systems currently on the market are the OSGI module system[1], as used by the Eclipse rich client platform (RCP) and integrated development environment (IDE)[2], and a custom module system that is used and promoted by the NetBeans RCP and IDE[3]. Both systems allow to restrict the visibility of classes inside a module to other modules, thereby promoting strong decoupling of the components of an application. Decoupling and strong encapsulation are generally regarded as good implementation patterns for larger software, where the communication between modules should be based on contracts, as defined by publicly available interfaces. The actual implementations of the interfaces remain private and thus unreachable for other modules. This strong restriction can be relaxed by introducing *friend* dependencies, that allow specifically defined modules to access otherwise hidden classes within a friend module.

Another aspect of both OSGI and the NetBeans module system are the explicit requirements for versioning of the modules. Versioning allows to explicitly check modules for their compatibility, since each module states the minimum required versions of other modules required by it to properly function. This also allows to safely update modules without breaking backwards compatibility, while still allowing to introduce new functionality that can be used by an updated dependent module.

We decided to use the NetBeans RCP due to its superior support for module creation and maintenance and the user-friendly management of module updates. The OSGI module system may be more powerful in some aspects, but nowadays NetBeans includes an OSGI-compatible runtime container to execute OSGI or Eclipse modules within NetBeans RCP applications. Additionally, NetBeans modules can expose OSGI information, so that components developed on either platform and module system can be deployed within the other system with ease. An example of an application in the domain of chromatography-mass spectrometry that is mainly used in the area of analytical chemistry and which is based on the OSGI-compatible EclipseRCP is OpenChrom (Wenig and Odermatt 2010) (see Section 3.1).

For Maui, the following requirements were defined, based on the aforementioned considerations:

- Modularity for easy extensibility and separation of concerns, realized by using the module system provided by the NetBeans RCP;
- Comprehensive visualization with context sensitive actions and cross-linking of data throughout the application;
- High interactivity, especially for large datasets and large samples;
- Simple integration of external tools;

1. http://www.osgi.org
2. http://eclipse.org
3. http://www.netbeans.org

- Integration of MALTCMS and of MALTCMS pipelines;
- Statistical analysis and visualizations.

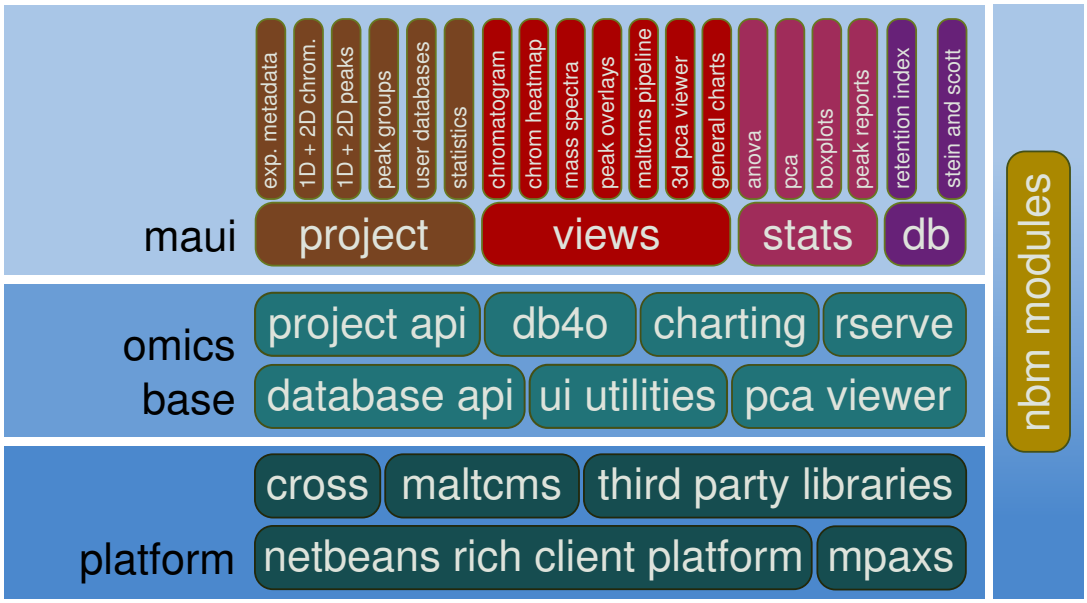


Figure 6.1.: Software layers and subsystems of MAUI.

A conceptual overview of MAUI's software layers and modules is depicted in Figure 6.1. The lowest level in this hierarchy is based on the CROSS, MALTCMS, MPAXS, and other associated third-party modules in NetBeans NBM format. These are further complemented by the infrastructure modules provided by the NetBeans RCP. They provide basic graphical user interface (GUI)-related modules for global action and selection management, coordinated start-up and shut down of the application, as well as the actual module system (the runtime container), and modules for persistent settings and views.

The next level consists of the *omics-base* cluster that contains modules that provide the common functionality between MAUI, and the gel-based proteomics application PROTEUS (unpublished, developed by Konstantin Otte). This layer provides a common project API, a generic database access API and a corresponding reference implementation for the object database db4o[4]. Additional functionality is provided for the domain object-specific association of user interface components (views), general interactive charting, backed by the data model, an interface to the statistical software R[5] via *Rserve*[6] and the JAVA-based 3D viewer for PCA results, realized by Leonhard Stutz originally for MeltDB (Kessler et al. 2013; Neuweger et al. 2008). We describe the top level layer of MAUI in the following sections.

4. http://www.db4o.com
5. http://www.r-project.org
6. http://rforge.net/Rserve

6.2. Project Model

MAUI's project model consists of two modules, one providing the API to the project domain model, the other providing the implementation of the project backed by a local db4o database.

The project API is modeled as a hierarchical tree structure, similar to a file system, where the project node represents the root of the tree. Below the tree, different container types hold data descriptors relevant to the project, such as sample and *treatment group* containers, *peak* and *peak group* containers, and *statistics* containers. These containers can contain other containers or individual descriptors, e.g. representing chromatograms, peaks, peak groups, or statistical results. The implementation is not specific to db4o and the data can easily be transferred to a hierarchical XML format for backup or migration purposes.

This structure also allows other modules to add containers and descriptors to the project database that are not already defined within the project implementation, and to provide appropriate actions and views for them. Thus, a loose coupling of modules with different responsibilities is achieved.

The project is also structured on the file system, containing folders for processing results, imported or created user databases with metabolite information, and a folder for custom scripts that extend MAUI's functionality (see Section 6.5.2).

6.2.1. Project Creation

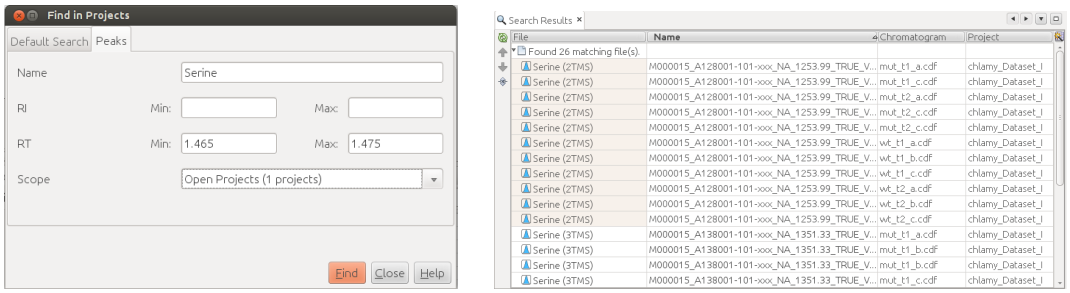
The process of project creation in MAUI is realized by a guided dialog, a *wizard*. The current wizard is specialized for metabolomics experiments where different treatment groups (factor combinations) and sample groups (technical replicates) can be specified. A treatment group can contain multiple sample groups, one for each biological specimen. Each sample group represents a biological replicate within its treatment group, while the members of a specific sample group represent technical replicates (repeated measurements of the same sample).

6.3. Data Import and Export

The integration of and interaction with other software tools is of crucial importance for a diverse field such as metabolomics. MAUI therefore supports a number of different methods for data import and result export.

6.3.1. Peak Import

To access *peak* data, MAUI provides support for peak reports exported by the LECO ChromaTOF software. It is possible to import both one- and two-dimensional peak data, either mapping the data onto existing chromatograms, or to directly create a project from them. Furthermore, MAUI can import one- and two-dimensional peak reports created by MALTCMS. Each imported peak report receives a unique identifier,



(a) Peak search dialog in MAUI. Peaks can be searched across open projects based on their name, their retention index and retention time ranges.

(b) Result view after peak search. The result after a peak search is presented in tabular format, stating the chromatogram descriptor parent of each peak and its containing project.

Figure 6.2.: Peak search dialog and result view in MAUI.

so that each chromatogram can have an arbitrary number of attached peak lists, e.g. to compare peak finding efficiency between different algorithms or parametrizations.

Peaks may be searched via the peak search dialog (see Figure 6.2(a)) across open projects and by different criteria. After the search criteria, like name, retention index range, and retention time range have been entered, the result is presented in tabular form stating for each peak in which project it is contained and to which sample it is associated (see Figure 6.2(b)). Peaks can be selected and the corresponding information is available to all other components via the selection system provided by the NetBeans platform.

6.3.2. Peak Group Import

In order to map peaks from different samples into *peak groups*, an external alignment file in the form of a CSV file, following the format of the *multiple-alignment.csv* file created by BIPACE and CEMAPP-DTW can be imported. Each row within that file format represents a distinct peak group, while a column represents the associated sample file name (without file extension). Each peak within a row is identified by its associated mass spectrum index (scan_index) within the raw file. Due to the modular structure, the mapping process can be customized to use other identifiers as well.

6.3.3. MeltDB

The web-based MeltDB LIMS and analysis system for metabolomics data (Kessler et al. 2013; Neuweger et al. 2008) can be accessed from within MAUI to allow limited bidirectional exchange of peak data. However, this requires that the raw data has been submitted to MeltDB and that a corresponding project exists. This is currently not possible for GC×GC-MS experiments due to size constraints in the upload form of MeltDB.

6.3.4. CSV Export

MAUI provides export of annotated peak groups to a custom CSV format that can be opened easily in common spreadsheet programs like Microsoft Excel for further downstream processing or combination with other tool results. The peak areas in these reports are normalized according to user settings for internal and external sample normalization. Furthermore, the annotated ANOVA peak groups can be exported together with the calculated p-values, F-values and degrees of freedom. The method used for multiple testing correction is reported as well. The report also includes putative peak identifications, corresponding majority group names, average retention times, and further information.

6.4. Visualization

Maui provides different visualizations depending on the type of the domain object in the current selection. Chromatograms with one separation dimension can be visualized as TIC chromatograms with interactive selection and highlighting of peak annotations and raw data. The mass spectra of selected scans are dynamically updated within the mass spectra view. Figure 6.3 provides an impression of the complete application, in this case for a project containing raw GC×GC-MS datasets and peak descriptors imported from ChromaTOF for CHLAMY Dataset I (see Section 4.4.3).

Panel (a) of Figure 6.3 contains the project explorer that shows the project's hierarchical structure with four treatment groups at the top. The sample node for mut_t1_a.cdf is expanded and shows the imported peak list container associated to the sample, labeled by the name of the tool that imported the peak list. Panel (b) is the welcome center component that provides a categorized, interactive guide to getting started with MAUI. Links within the component directly invoke the appropriate system action so that novice users do not need to known the complete menu hierarchy to get started. Panel (c) shows the heatmap view for two-dimensional chromatography data. The color scheme used for rendering can be selected and customized via the 'Paint Scale' button. The 'Range' slider allows to restrict the displayed range of values and thus can be used to improve the visualization. In Panel (d), a mass spectrum selected within the chromatogram heatmap view is shown. The selection is reflected in the context-aware navigator (e), that manages the currently active selection for the view component that currently holds the focus within the application.

Panel (f) in the middle of the application window shows the same chromatogram as panel (c), but as a one-dimensional chromatogram, also with superimposed peak markers. Upon selection of a scan in the panel, the next peak descriptor is automatically added to the selection along with the raw data mass spectrum and the positions of both mass spectrum and peak descriptor are highlighted. The selection is reflected in the navigator if (f) is selected by clicking on it. The selections of panels (c) and (f)

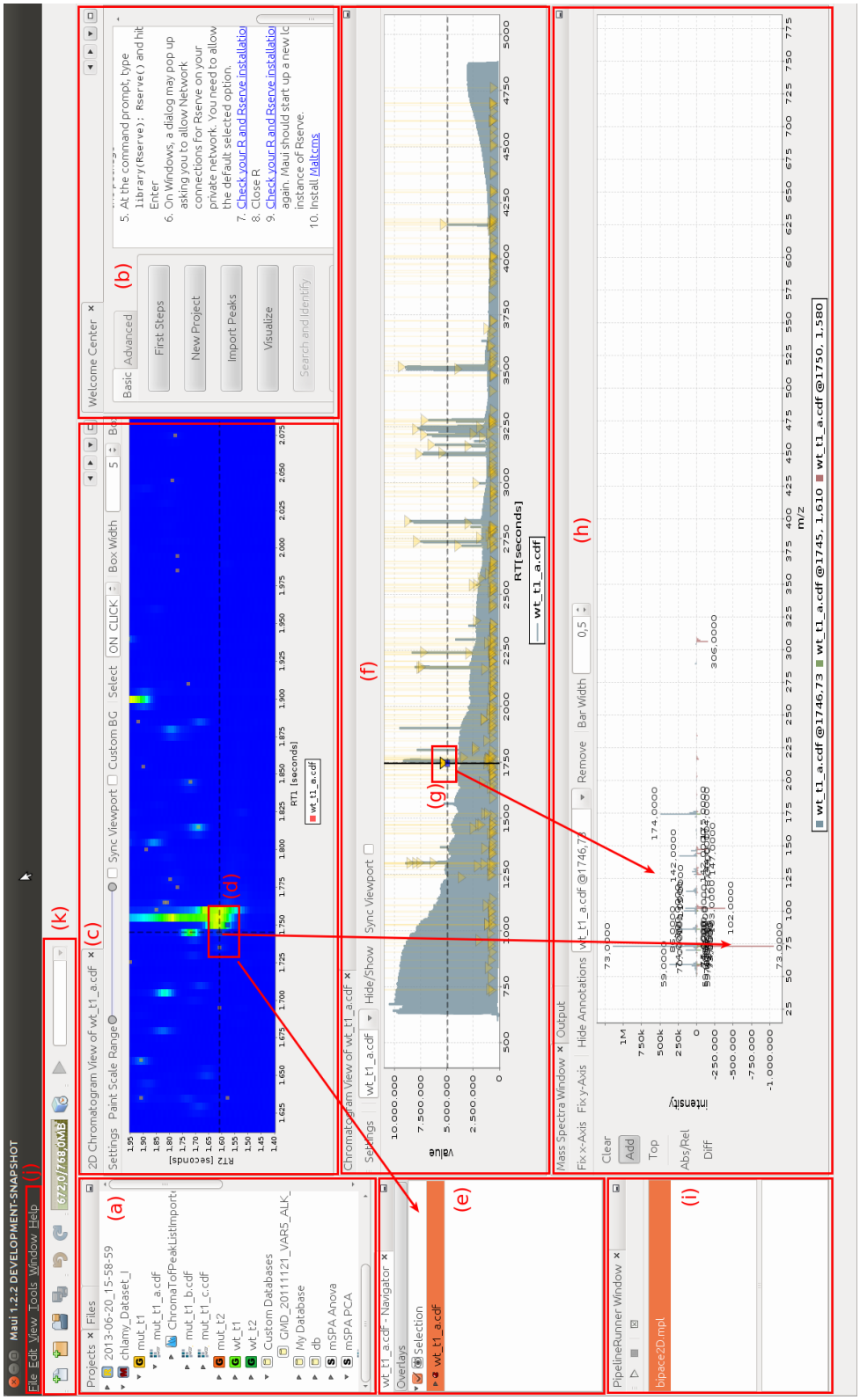


Figure 6.3.: Screenshot of the MAUI application showing the CHLAMY Dataset I from section C.5.

have been successively added to the mass spectrum panel (h) at the bottom of the application window. The mass spectrum panel supports different modes to react on selection events occurring in other panels.

Panel (i) in the lower left of the application window shows the pipeline runner component that is used to run MaltCMS processing pipelines. Pipeline configurations can be imported for a project and are editable from within the application (see Figure 6.4(b)).

Finally, (j) is the main menu bar of the application that contains menu entries for the update system and for common tasks and views. The toolbar (k) contains shortcuts to create new files and projects, and to open existing projects.

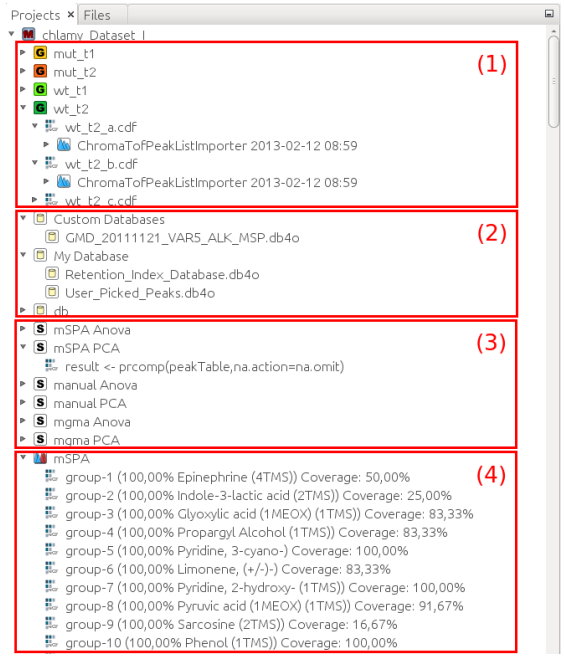
6.4.1. Project Explorer

Figure 6.4(a) shows an expanded view of the project explorer. In panel (1), the treatment group containers are visible, for the four distinct factor combinations of wild type (wt), mutant (mut), time point 1 (t1) and time point 2 (t2). Is is possible to assign a distinct color to each group, which is also used by the chromatogram and peak group visualizations to better distinguish between them. Panel (2) just below shows custom database containers that allow the grouping of different databases that can be used for putative peak identification or for the calculation of retention indices. Databases can be created directly from existing peak annotations. The next panel (3) shows different statistics containers, in this case for the different reference multiple alignments that were used in the evaluation of BiPACE 2D (see Section 4.4.3). For each reference, the view shows statistical descriptors for ANOVA and for PCA, which were each calculated using the *Rserve* backend in Maui. Panel (4) finally shows an aligned peak group, in this case derived from the reference multiple alignment generated by the mSPA method. Each group descriptor immediately shows how many of the peaks within it were annotated with the majority name of the group and how many samples the peak group covers (coverage). In this example, groups 5, 7, and 10 cover all samples and all peaks within the groups have identical putative identifications.

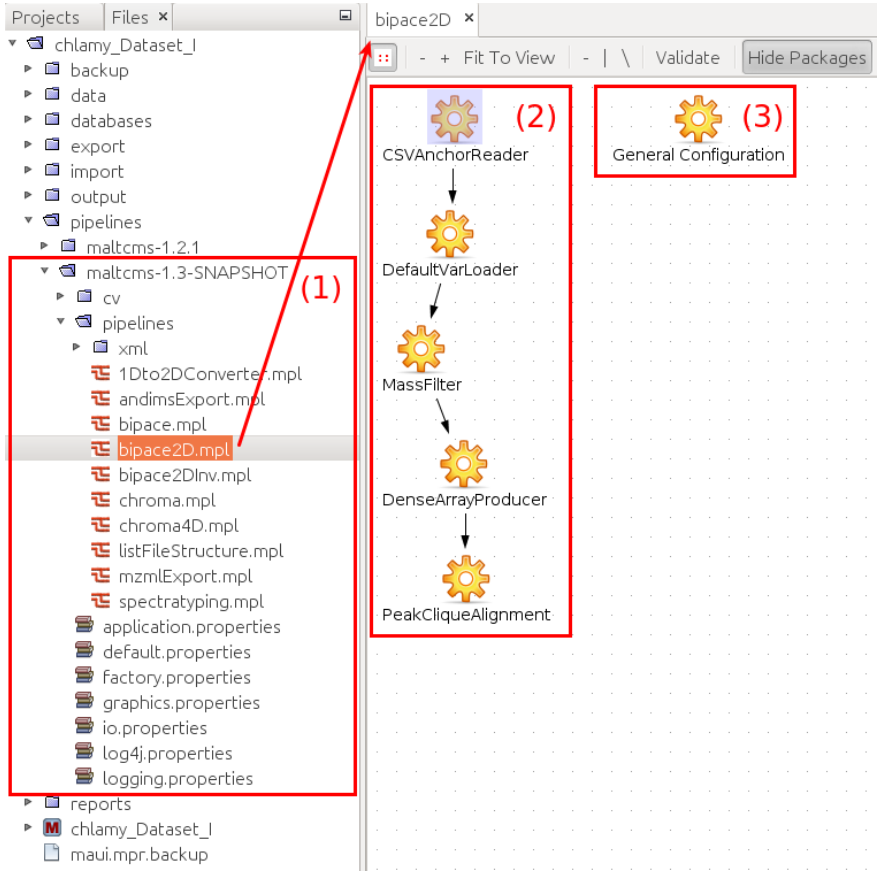
The project explorer view also provides an alternative view of the actual files and folders contained below the project location (see Figure 6.4(b)). The 'pipelines' folder contains imported configurations of MaltCMS pipelines (1). The corresponding files can be opened and edited in Maui. The actual elements of the linear pipeline (2) can be customized individually. The general configuration of the pipeline (3) can also be customized directly from within the editor. It is also possible to open and edit the pipeline configuration and the corresponding XML file within a standard text editor.

6.4.2. Chromatogram Views

We already showed two different chromatogram views in Figure 6.3. However, Maui does not impose a principal limit on the number of open chromatogram views.



(a) Explorer view of the project tree of CHLAMY Dataset I in MAUI.



(b) Explorer view of the file hierarchy of CHLAMY Dataset I in MAUI and the visual editor of the selected MALTCMS pipeline.

Figure 6.4.: Explorer views of the project and file tree of CHLAMY Dataset I in MAUI.

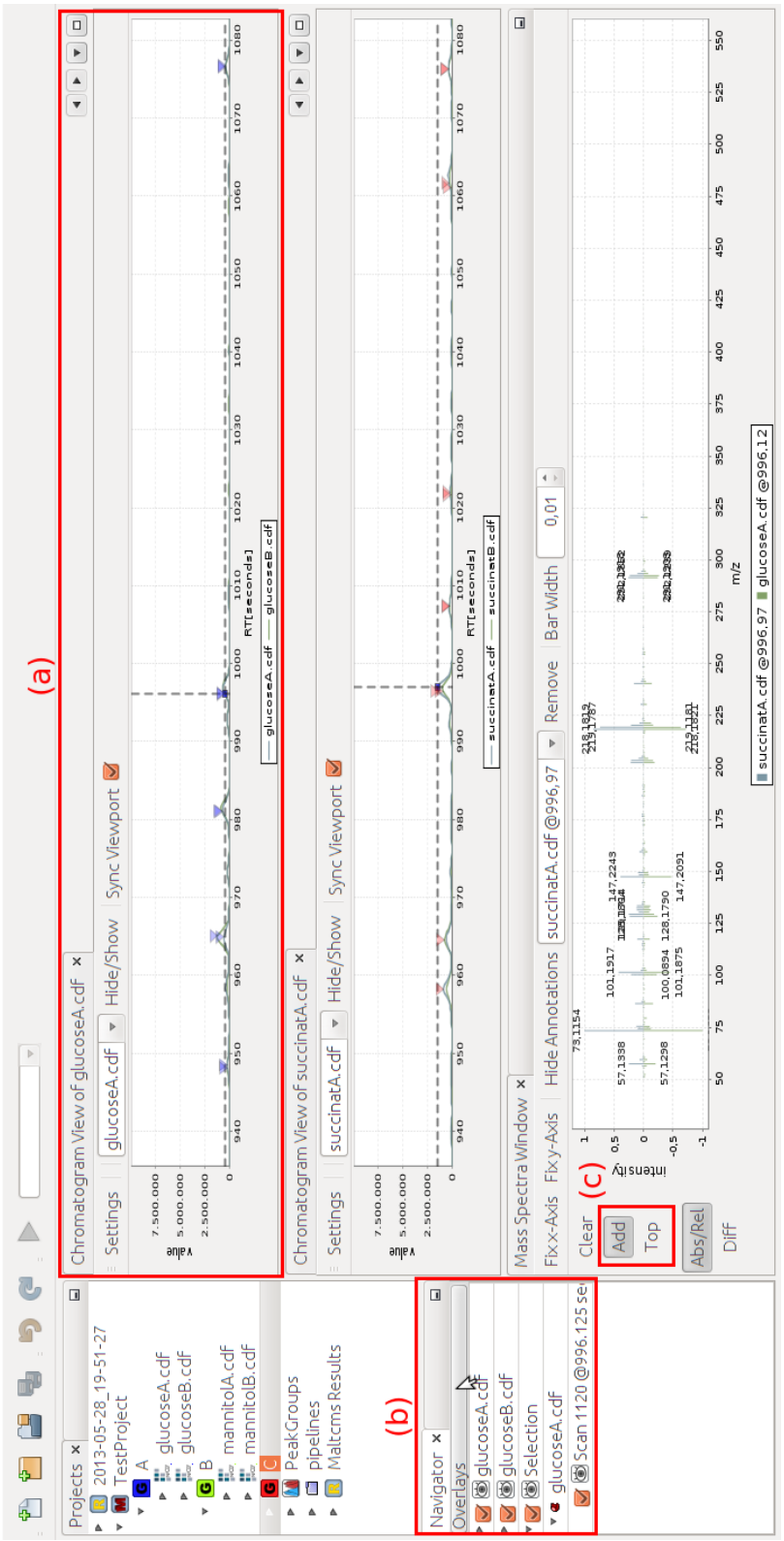


Figure 6.5.: Synchronized TIC view for samples from two different sample groups. Visible retention time ranges are synchronized across the individual views. Individual mass spectra can be selected distinctly for each chromatogram in the views.

Figure 6.5 shows two separate one-dimensional chromatogram views (panel (a) and the view below it). These views can show an arbitrary number of chromatograms simultaneously. In this case, each view shows the chromatograms of one sample group. The views can be synchronized ('Sync Viewport'), so that if the viewport of one view is changed by the user, e.g. by zooming or panning the view, the other view is updated to show the same viewport. Selection of mass spectra and peaks is performed individually for each chromatogram within the views and the selected peak descriptors and raw mass spectra are visible in the navigator view component (b). The visibility of the selection, as well as of the peak descriptors, can be changed from within the navigator.

Finally, the selection management allows to compare selected peak descriptors, mass spectra, or complete peak groups within the mass spectrum view component (c). This view is multi-modal. In its default mode of operation, only the objects contained in the currently active selection are shown in it. If the 'Add' button is toggled, the user can add the next selected object to the mass spectrum view. By changing the 'Top' button, the new mass spectrum can be added below the top one, for better visual comparability. Additionally, the view allows to toggle between absolute and relative intensity scaling of the mass spectra via the 'Abs/Rel' button. If only two mass spectra are shown, the 'Diff' button toggles a difference view that immediately shows which m/z values differ between them.

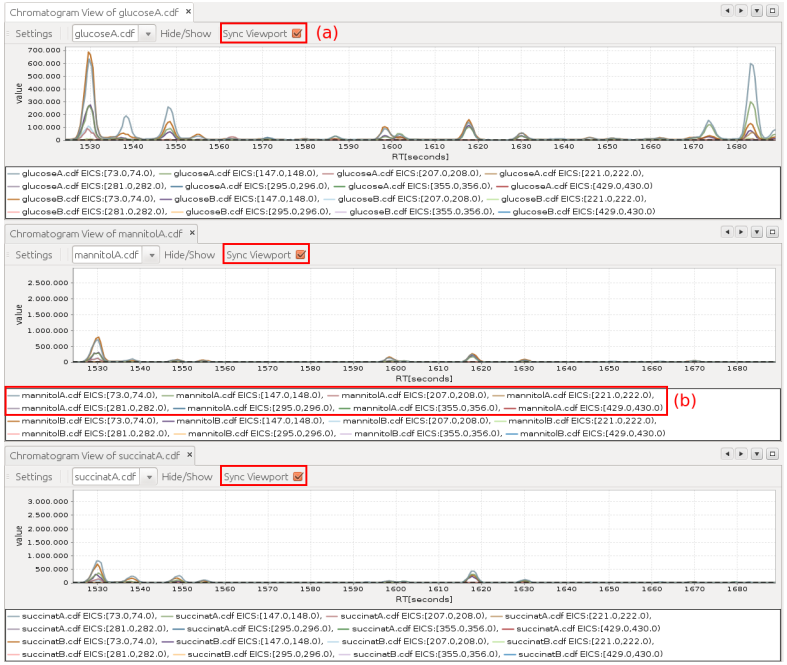


Figure 6.6.: Synchronized EIC view for samples from three different sample groups.

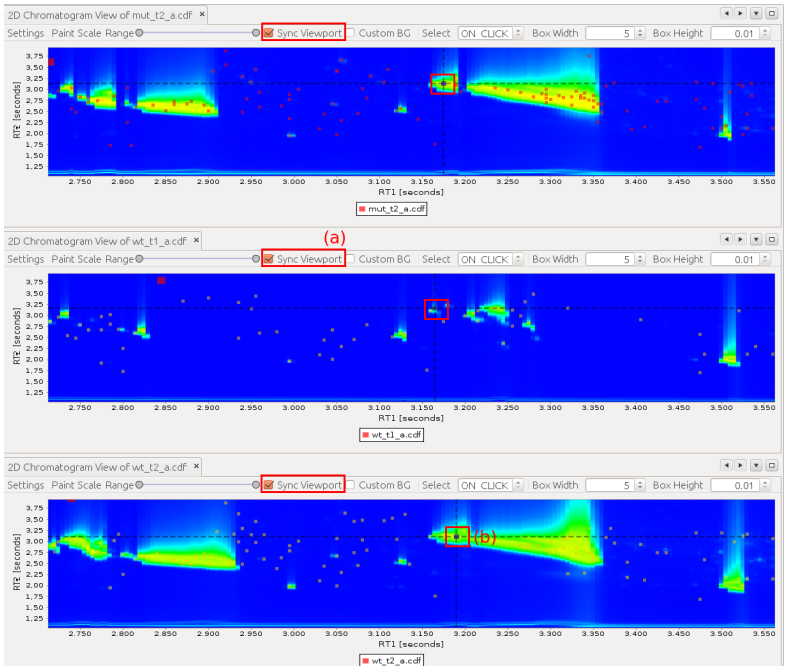


Figure 6.7.: Maui 2D chromatogram view with overlayed peak markers imported from a ChromaTOF peak report.

The one-dimensional chromatogram views also support a different mode of operation that is shown in Figure 6.6. Again, it is possible to synchronize the view ports of multiple instances of the chromatogram viewer (a), in this case each for a separate treatment group (glucose, mannitol, succinat). Via the 'Settings' button, the views can be changed individually to display a subset of user-defined EICs, either as separate series for each EIC (b), or in summation mode to display a summed ion current. This mode is especially useful to compare the elution profiles of known or suspected ion m/z values. Again, each series can be selected individually and the corresponding full mass spectrum is selected and visualized in the mass spectrum viewer (not shown).

The two-dimensional chromatogram views shown in Figure 6.7 illustrate that their viewports can also be synchronized (a). Peak selection is reflected only in the currently focused view component (b) and the selection is again managed in the navigator (not shown) and mass spectrum views. For fast browsing of mass spectra, one can switch the selection mode from ON_CLICK to ON_HOVER, so that the selection is updated while moving the mouse over the chromatogram. The hover-selection runs fluently even on GC×GC-MS raw files exceeding sizes of 6 GB. The visualization can be customized with a user-selectable color gradient, background color and displayed color range.

6.4.3. Peak Group Views

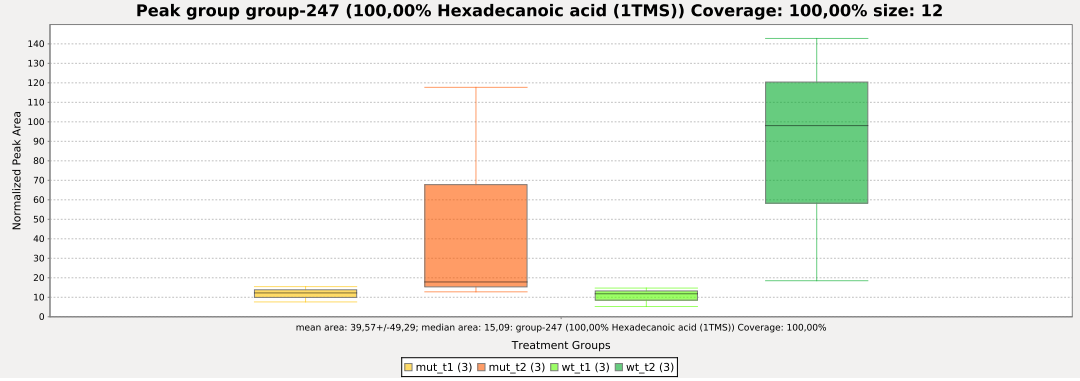


Figure 6.8.: Peak area boxplot for a peak group with significant fold changes between factor combinations MUT-T2 and WT-T2 from CHLAMY Dataset I (see Section 4.4.3). Normalization is based on the peak area of the Ribitol peak found in all samples.

Peak groups in MAUI can be visualized based on the (normalized) area of each peak in an area boxplot that is partitioned and colored according to the treatment group of each peak's originating sample. Figure 6.8 shows such a box plot for a peak group that showed significant differences in average peak area between different factor combinations. The plot immediately shows the coverage of the peak group and the putative identification. Plots like this one can be exported as scalable vector

graphics (SVG) for subsequent editing and conversion to the portable document format (PDF), e.g. for publication purposes.

Alternatively, the peak retention times can be visualized as a boxplot, much like the area box plot that we just described. This view is especially useful to check whether the assigned peak groups are plausible. High deviations in retention times may be a hint towards misalignment of the peaks that are part of the visualized peak group.

6.5. Data Processing

MAUI provides different possibilities for the processing of data from metabolomics experiments following the steps of the general pipeline defined in Section 2.6. Generally, modules can provide their own custom functionality based on the project API objects, integrating directly with MAUI and other modules. A second approach, taken by the MALTCMS integration is to control external tools from within MAUI and to provide facilities to import their processing results. The third approach is provided by the scripting integration via Groovy, which can combine aspects of the previous two, by having access to the MAUI API objects directly, but at the same time allowing easy integration of external tools via a simple scripting interface.

6.5.1. Maltcms Integration

MAUI's support for MALTCMS allows to download the most recent development or stable versions from the project website. The currently active version of MALTCMS can be set from within the options dialog for all open projects. Within the projects, pipelines from the active version can be imported and are directly visible in the project's context menu for execution. After execution of a pipeline on the project data, MAUI provides different actions to selectively import MALTCMS processing results, like peak lists and multiple peak alignments.

6.5.2. Scripting

MAUI supports scripting actions in the Groovy programming language[7]. Groovy is fully compatible with JAVA and easy to integrate. MAUI provides template groovy scripts for the easy creation of actions on the project, chromatograms, and peak group objects. The scripts have full access to MAUI's project API and can therefore implement any desired functionality. It is hence easily possible to extend MAUI with custom code that should not be contained in a large-scale module or that is used for rapid prototyping purposes. We have integrated the matched filter peak finding method provided by XCMS (Smith et al. 2006) as a sample script within MAUI to demonstrate the integration of external tools.

7. http://groovy.codehaus.org

6.6. Statistical Evaluation

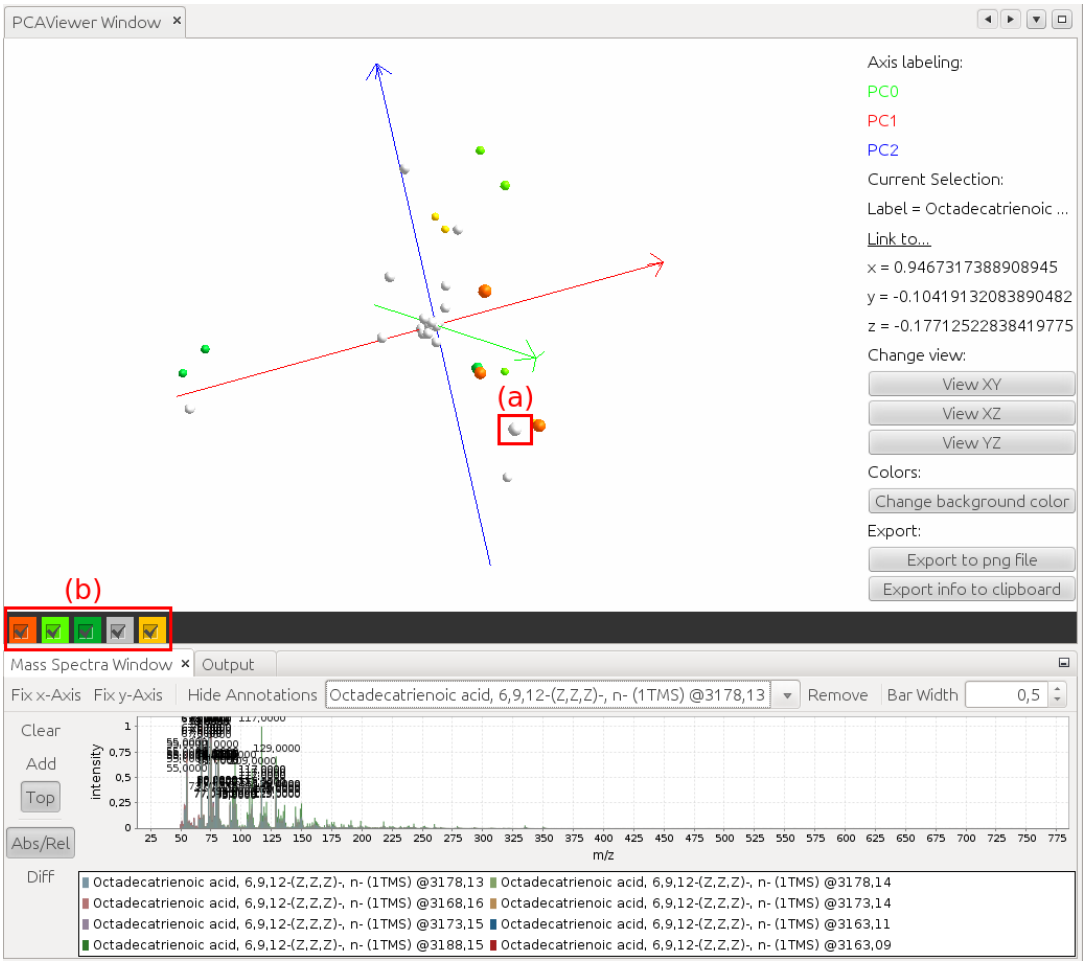


Figure 6.9.: MAUI 3D PCA view. PCA results calculated from aligned peak groups can be interactively visualized, with projected loadings for inidividual peak groups (white spheres) and individual samples (colored spheres). The group color corresponds to the group color assigned in the project. Selection of the peak group within the 3D view (a) is exposed to the selection lookup of the application, so that other components can react to the selection. Here, the mass spectrum view is updated according to the selection. Components of the plot can be hidden or shown individually (b).

The statistical backend of Maui is based on GNU R and the TCP/IP-based *Rserve* server. Currently, ANOVA with multiple testing correction, and PCA are available as statistical methods.

Figure 6.9 shows the visualization of a PCA descriptor that was calculated for the manually annotated reference multiple alignment from the CHLAMY Dataset I mentioned in Section 4.4.3 using the *Rserve* backend. The view is freely rotatable and zoomable. Colored spheres correspond to individual samples, while white spheres

correspond to the peak groups. The 3D view (a) highlights the currently selected peak group. The selection is mirrored within the mass spectrum view, showing the associated mass spectra of all members of the peak group simultaneously. The visibility of each sample and of the peak groups can be toggled individually as in (b). The current view can be saved as a PNG graphics file or to the system clipboard for editing with other software tools.

6.7. Peak Identification

External databases in MSP format can be imported into a project using MAUI. The user interface allows to classify the database as a user or retention index database and also provides a convenient view of all metabolites, their meta-data and mass spectra. The view also allows in-place editing of the metabolite entries, as well as inspection of their mass spectra and the comparison to mass spectra from imported peak lists or selected raw spectra from chromatogram views. Custom user databases can also be directly created within the project to manually create a reference database from selected peaks.

MAUI further provides a customizable search dialog to annotate all or a selected peak list within a project against an arbitrary number of selected databases. If available, a database classified as a RI database can be used to calculate RIs following the method of Den Dool and Kratz (1963). Using RIs can drastically improve the true positive rate of mass spectral identification against a database and also accelerates the database search, as only candidate metabolites within a small range around the calculated RI value need to be compared against the query. MAUI allows to define an RI window for the database search, if a suitable RI database is selected.

The mass spectral similarities available for database search are the weighted and plain cosine scores already mentioned in Section 3.3, but other similarities can easily be added through the module system.

7

Summary and Outlook

Metabolomics is still a field in its infancy when compared to established fields like genomics and also proteomics. It is therefore also rapidly growing, both in absolute numbers concerning the number of papers published within a metabolomics context, and technologically, where virtually every year manufacturers present new analytical machinery with improved mass resolution and accuracy, higher scanning speed and improved chromatographic separation. With new machinery available however, the amount of data acquired and in need of processing grows at an enormous pace. With large-scale projects like population cohort studies (Moayyeri et al. 2013) becoming more frequent, the need arises for customized and scalable software and at the same time both well-documented and publicly available algorithms.

In this thesis, we presented an overview of Open Source software that is available for the processing and analysis of GC-MS and GC×GC-MS data in different steps of a typical metabolomics workflow. We specifically described the requirements for automatic peak matching and alignment among multiple samples and presented algorithms that address the peak and chromatogram alignment problems for GC-MS and GC×GC-MS chromatograms. We have described BiPACE in Chapter 3, a novel algorithm based on a suitable pairwise peak similarity and a graph theoretic approach to identify conserved and reliable cliques of peaks throughout different chromatograms without requiring a reference chromatogram. BiPACE was evaluated on its own and in conjunction with the DTW-based chromatogram alignment algorithm CeMAPP-DTW against a manual reference multiple alignment and against a larger reference multiple alignment that was generated using MeltDB (Kessler et al. 2013; Neuweger et al. 2008). The results were good and showed that especially BiPACE reported conservative alignments with few false positives. CeMAPP-DTW was demonstrated to work well in combination with BiPACE, avoiding most of the overfitting that is often associated with DTW (Hoffmann et al. 2012). Nonetheless, even though it uses considerably less memory and computational time than plain DTW, CeMAPP-DTW is still expensive to compute. An alternative is to apply BiPACE to all mass spectra of the chromatograms, while using a

maximum retention time difference window to avoid the comparison of remote peak mass spectra.

For the peak alignment problem for GC×GC-MS chromatograms, we developed the BiPACE with two-dimensional retention time (BiPACE 2D) algorithm as an extension of BiPACE, with a specialized peak similarity function handling two-dimensional retention times. We evaluated BiPACE 2D and its relatives on four diverse datasets and compared them against other publicly available algorithms with superior results on three of the four datasets. BiPACE 2D performed well due to its low false positive rate and generally high true positive rate in comparison to the other methods. The dataset where BiPACE 2D did not perform better than the other algorithms was acquired under varying temperature gradient conditions. It is therefore not representative of typical metabolomics experiments, where the temperature gradient is usually determined in advance to optimize the chromatographic separation, and all subsequent samples are then acquired under identical conditions. The parameterization of BiPACE 2D requires some domain knowledge in advance, e.g. the expected standard deviations in the first and second retention time dimensions, and some fine-tuning of the threshold levels for the retention time deviations. The only other important parameter is the minimum clique size. We showed, that the highest number of true positives is achieved with a minimum clique size of 2, requiring that a clique covers at least two chromatograms. This also includes all cliques with a better support over a larger number of chromatograms, thus, this parameter only influences the reporting of cliques, not the clique-finding itself.

BiPACE 2D was faster and required less memory than any of the other methods, but this does not hold true for much larger sample sizes and peak numbers. There, the quadratic runtime complexity, both in the number of samples and in the number of peaks, slows BiPACE 2D down in comparison to the other methods. This can be alleviated by the parallelization of the pairwise similarity calculation and BBH determination phases of the algorithm. One future improvement of BiPACE and BiPACE 2D would thus be the parallelization of the clique finding and merging phase by exploiting the *k*-partite properties of the graph, applying the ideas laid out by Schmidt et al. (2009) to the restricted *k*-partite graph that BiPACE uses. Additionally, the merging phase is partially order-dependent, if cliques share common peaks, which may pose additional problems during parallelization. This can be relieved by the BBH percentage criterion that also allows non-fully connected subgraphs to be counted as cliques. Usually, only few cliques have such merge conflicts, so that a different merging strategy than the current greedy approach could improve on the already good results of the algorithm. It is also worth to consider whether parts of the adjacency lists used to represent the connectivity of the graph could better be stored outside of main memory in order to allow even larger data sets to be processed. This could be realized by using a graph database like Neo4j[1], which also allows for parallel access and processing.

1. http://www.neo4j.org

We further addressed the peak finding and integration problems for GC×GC-MS data by the CWT peak finder and the SRG peak integration methods, described in Chapters 4 and 5. These methods seem promising in both their execution speed and in their preliminary results. However, they still lack a thorough evaluation against other methods, most of which are proprietary methods only available in commercial software like ChromaTOF and GCImage. Their benefit is that they are trivially parallelizable and thus can be scaled almost linearly with the number of chromatograms on a suitable computing grid infrastructure.

The methods that we presented in this thesis have been implemented in the Open Source framework MALTCMS and are available to the public without cost. MALTCMS and its underlying frameworks CROSS and MPAXS, together with a host of accompanying Open Source libraries, enable other researchers to quickly customize their own processing workflows. Even the integration of additional analytical methods is possible since the primary data abstraction used by CROSS is generic and extensible enough to support additional formats and methods. Thus, MALTCMS can also be seen as a generic technology integration platform and not only as a specialized framework for the processing of GC-MS, GC-FID (see Appendix A.3), and GC×GC-MS data. Lately, the MALTCMS framework was extended to support LC×LC-MS as an analytical platform, also providing support to access MS data at different fragmentation levels from MS^N acquisitions. One advantage of MALTCMS over similar frameworks like MZMINE 2 or GUINEU is that developers can easily choose which modules they want to use and that it is independent of a GUI.

We described MAUI, the GUI for MALTCMS in Chapter 6. It provides a convenient and extensible project model, supplemented with metadata for experiment grouping, conditions and additional information, that can be used as a basis for the creation, configuration, and execution of MALTCMS workflows. MAUI is easy to customize and can be augmented easily with additional functionality through user modules. It is based on an industry standard module system and adheres to the conventions of the platform it was developed on. It is therefore possible to install all modules of MAUI into the regular NetBeans IDE and use MAUI modules and functionality alongside regular software projects. We think that especially this feature makes it a good candidate for projects that want to implement and provide customized and extended functionality, and this also makes it amenable especially to bioinformaticians and domain specialists (see Appendix A.4 for an example). However, in their present state, both MALTCMS and MAUI cover only some parts of the typical metabolomics pipeline and need additional functionality that is provided by other software, e.g. *Rserve*, XCMS, and ChromaTOF. Further improvements include the development and integration of methods for quality control of large sample batches in both MALTCMS and in MAUI to enable higher levels of automation in large scale studies.

7.1. Future Directions

The present state of Open Source frameworks for metabolomics is very diverse. A number of tools have seen steady development and improvement over the last years, such as XCMS, MzMine 2, and PyMS, while others are still being developed, such as MzMatch, Guineu, and Maltcms. There is currently no framework available that covers every aspect of metabolomics data preprocessing. Most of the frameworks concentrate on one or a few analytical technologies with the largest distinction being between GC-MS, LC-MS, and NMR. GC×GC-MS raw data processing is currently only handled by Maltcms' ChromA4D pipeline and by Maui, while Guineu processes peak lists exported from LECO's ChromaTOF software and offers statistical methods for sample comparison together with a user-friendly graphical interface.

Since metabolomics is an evolving field of research, no framework captures all possible use-cases, but it will be interesting to see which frameworks will be flexible and extendable enough to be adapted to new requirements in the near future and whether there will be a convergence of functionality between them. The Eclipse Science Industry Working Group (IWG)[2] may be a possible third-party organization that allows different frameworks to retain their identity towards the community, but at the same time provides the necessary infrastructure that allows to share common functionality which is used and required by virtually all scientific projects. The IWG currently contains a wide variety of projects from physics, chemistry, biology, bioinformatics, and medicine. Maltcms and OpenChrom are part of the IWG and have already profited from mutual exchange of ideas and best practices.

In order to combine experiments from multiple *omics* experiments, another level of abstraction on top of local or web-service based tools for data processing, fusion, and integration of metabolomics experiments is a necessary future requirement. Generic workflow systems like Knime (Berthold et al. 2009), Taverna (Hull et al. 2006) or Conveyor (Linke, Giegerich, and Goesmann 2011) offer integration of such resources, augmented with graphical editors for *point-and-click* user interaction. However, due to their generic nature these systems are far away from being as user-friendly as applications designed for a specific data analysis task and require some expert knowledge when assembling task-specific processing graphs.

One point that requires further attention is the definition and controlled evolution of peak data formats for metabolomics, along with other formats for easier exchange of secondary data between applications and frameworks. A first step in this direction has been taken by Scheltema et al. (2011) by defining the PeakML format. However, it is important that such formats are curated and evolved, possibly by a larger non-profit organization like the Human Proteome Organization (HUPO) within its proteomics standards initiative (PSI). Primary data is already acessible in a variety of different, defined formats, the most recent addition being mzML (Martens et al. 2011) which is curated by the PSI. Lately, mzML has been extended to handle terms from analytical technologies that are mainly used in metabolomics research. The mzQuantML and

2. http://wiki.eclipse.org/Science_IWG

mzTab initiatives try to establish comparable standards for secondary qualitative and quantitative peak data (E. W. Deutsch 2012). To improve interoperability with other open software, the realization of mzTab support for MaltCMS and Maui, both as input and output formats, should be addressed in the near future, once the standard for mzTab in metabolomics has been finalized.

We think that the main contributions of this thesis, MaltCMS and Maui, provide a good starting point for customized metabolomics workflows and applications and we hope that they will continue to prove themselves as valuable, developer-oriented tools in computational metabolomics.

Bibliography

Åberg, K. Magnus, Erik Alm, and Ralf J. O. Torgrip. 2009. "The correspondence problem for metabonomics datasets." *Analytical and Bioanalytical Chemistry* 394, no. 1 (May): 151–162.

Adams, Rolf, and Leanne Bischof. 1994. "Seeded region growing." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16 (6): 641–647.

Ahmad, Isthiaq, Frank Suits, Berend Hoekman, Morris A. Swertz, Heorhiy Byelas, Martijn Dijkstra, Rob Hooft, Dmitry Katsubo, Bas van Breukelen, Rainer Bischoff, and Peter Horvatovich. 2011. "A high-throughput processing service for retention time alignment of complex proteomics and metabolomics LC-MS data." *Bioinformatics* 27, no. 8 (April): 1176–1178.

Aluru, Srinivas, and Fatih E. Sevilgen. 1999. "Dynamic Compressed Hyperoctrees with Application to the N-body Problem." In *Foundations of Software Technology and Theoretical Computer Science,* edited by C. Pandu Rangan, V. Raman, and R. Ramanujam, 21–33. Lecture Notes in Computer Science 1738. Berlin, Heidelberg: Springer, January.

Amador-Muñoz, Omar, and Philip J. Marriott. 2008. "Quantification in comprehensive two-dimensional gas chromatography and a model of quantification based on selected summed modulated peaks." *Journal of Chromatography A* 1184 (1–2): 323–340.

Arnaud, M. J., A. Thelin-Doerner, E. Ravussin, and K. J. Acheson. 1980. "Study of the demethylation of [1,3,7-Me-13C] caffeine in man using respiratory exchange measurements." *Biological Mass Spectrometry* 7 (11-12): 521–524.

Aura, Anna-Marja, Ismo Mattila, Tuulikki Seppänen-Laakso, Jarkko Miettinen, Kirsi-Marja Oksman-Caldentey, and Matej Orešič. 2008. "Microbial metabolism of catechin stereoisomers by human faecal microbiota: Comparison of targeted analysis and a non-targeted metabolomics method." *Phytochemistry Letters* 1, no. 1 (April): 18–22.

Babushok, V. I., P. J. Linstrom, J. J. Reed, I. G. Zenkevich, R. L. Brown, W. G. Mallard, and S. E. Stein. 2007. "Development of a database of gas chromatographic retention properties of organic compounds." *Journal of Chromatography A* 1157, nos. 1-2 (July): 414–421.

Baldwin, J. E., and H. Krebs. 1981. "The evolution of metabolic cycles." *Nature* 291, no. 5814 (June): 381–382.

Baraldi, Eugenio, Silvia Carraro, Giuseppe Giordano, Fabiano Reniero, Giorgio Perilongo, and Franco Zacchello. 2009. "Metabolomics: moving towards personalized medicine." *Italian Journal of Pediatrics* 35, no. 1 (October): 30.

Barsch, Aiko, Thomas Patschkowski, and Karsten Niehaus. 2004. "Comprehensive metabolite profiling of *Sinorhizobium meliloti* using gas chromatography-mass spectrometry." *Functional & Integrative Genomics* 4, no. 4 (October): 219–230.

Bassham, James A., Andrew A. Benson, and Melvin Calvin. 1950. "The Path of Carbon in Photosynthesis VIII. the Role of Malic Acid." *Journal of Biological Chemistry* 185, no. 2 (August): 781–787.

Benfenati, E., P. Tremolada, L. Chiappetta, R. Frassanito, G. Bassi, N. Di Toro, R. Fanelli, and G. Stella. 1990. "Simultaneous analysis of 50 pesticides in water samples by solid phase extraction and GC-MS." *Chemosphere* 21 (12): 1411–1421.

Berg, Mark de. 2000. "Chapter 14 - Quadtrees." In *Computational Geometry: Algorithms and Applications,* 2nd ed. Berlin, Heidelberg: Springer.

Berk, Maurice, Timothy Ebbels, and Giovanni Montana. 2011. "A statistical framework for biomarker discovery in metabolomic time course data." *Bioinformatics* 27, no. 14 (July): 1979–1985.

Berthold, Michael R., Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. 2009. "KNIME - the Konstanz Information Miner: Version 2.0 and Beyond." *SIGKDD Explor. Newsl.* 11, no. 1 (November): 26–31.

Biller, J. E., and K. Biemann. 1974. "Reconstructed Mass Spectra, A Novel Approach for the Utilization of Gas Chromatograph-Mass Spectrometer Data." *Analytical Letters* 7 (7): 515–528.

Black, Robin M., Raymond J. Clarke, Robert W. Read, and Michael T. J. Reid. 1994. "Application of gas chromatography-mass spectrometry and gas chromatography-tandem mass spectrometry to the analysis of chemical warfare samples, found to contain residues of the nerve agent sarin, sulphur mustard and their degradation products." *Journal of Chromatography A* 662, no. 2 (February): 301–321.

Brasseur, Catherine, Jessica Dekeirsschieter, Eline M.J. Schotsmans, Sjaak de Koning, Andrew S. Wilson, Eric Haubruge, and Jean-Francois Focant. 2012. "Comprehensive two-dimensional gas chromatography–time-of-flight mass spectrometry for the forensic study of cadaveric volatile organic compounds released in soil by buried decaying pig carcasses." *Journal of Chromatography A* 1255 (September): 163–170.

Bron, Coen, and Joep Kerbosch. 1973. "Algorithm 457: Finding all cliques of an undirected graph." *Commun. ACM* 16, no. 9 (September): 575–577.

Bylund, Dan, Rolf Danielsson, Gunnar Malmquist, and Karin E. Markides. 2002. "Chromatographic alignment by warping and dynamic programming as a pre-processing tool for PARAFAC modelling of liquid chromatography–mass spectrometry data." *Journal of Chromatography A* 961, no. 2 (July): 237–244.

Callaghan, Sean O', David P. De Souza, Dedreia L. Tull, Ute Roessner, Antony Bacic, Malcolm J. McConville, and Vladimir A. Likić. 2010. "Application and comparative study of PyMS Python toolkit for processing of gas chromatography-mass spectrometry (GC-MS) data." In *Australasian Symposium on Metabolomics (2nd).* Melbourne 2010, October.

Callaghan, Sean O', David P. DeSouza, Andrew Isaac, Qiao Wang, Luke Hodkinson, Moshe Olshansky, Tim Erwin, Bill Appelbe, Dedreia L. Tull, Ute Roessner, Antony Bacic, Malcolm J. McConville, and Vladimir A. Likic. 2012. "PyMS: a Python toolkit for processing of gas chromatography–mass spectrometry (GC-MS) data. Application and comparative study of selected tools." *BMC Bioinformatics* 13, no. 1 (May): 115.

Carroll, Adam, Murray Badger, and A. Harvey Millar. 2010. "The MetabolomeExpress Project: enabling web-based processing, analysis and transparent dissemination of GC/MS metabolomics datasets." *BMC Bioinformatics* 11, no. 1 (July): 376.

Caspi, Ron, Tomer Altman, Kate Dreher, Carol A. Fulcher, Pallavi Subhraveti, Ingrid M. Keseler, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Lukas A. Mueller, Quang Ong, Suzanne Paley, Anuradha Pujar, Alexander G. Shearer, Michael Travers, Deepika Weerasinghe, Peifen Zhang, and Peter D. Karp. 2012. "The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases." *Nucleic Acids Research* 40, no. D1 (January): D742–D753.

Castillo, Sandra, Peddinti Gopalacharyulu, Laxman Yetukuri, and Matej Orešič. 2011. "Algorithms and tools for the preprocessing of LC-MS metabolomics data." *Chemometrics and Intelligent Laboratory Systems* 108, no. 1 (August): 23–32.

Chae, Minho, Robert Reis, and John Thaden. 2008. "An iterative block-shifting approach to retention time alignment that preserves the shape and area of gas chromatography-mass spectrometry peaks." *BMC Bioinformatics* 9, no. Suppl 9 (August): S15.

Christin, Christin, Huub C. J. Hoefsloot, Age K. Smilde, Frank Suits, Rainer Bischoff, and Peter L. Horvatovich. 2010. "Time Alignment Algorithms Based on Selected Mass Traces for Complex LC-MS Data." *Journal of Proteome Research* 9 (3): 1483–1495.

Christin, Christin, Age K. Smilde, Huub C. J. Hoefsloot, Frank Suits, Rainer Bischoff, and Peter L. Horvatovich. 2008. "Optimized Time Alignment Algorithm for LC-MS Data: Correlation Optimized Warping Using Component Detection Algorithm-Selected Mass Chromatograms." *Analytical Chemistry* 80, no. 18 (September): 7012–7021.

Clifford, David, Glenn Stone, Ivan Montoliu, Serge Rezzi, François-Pierre Martin, Philippe Guy, Stephen Bruce, and Sunil Kochhar. 2009. "Alignment Using Variable Penalty Dynamic Time Warping." *Analytical Chemistry* 81, no. 3 (February): 1000–1007.

Clote, Peter, and Jürg Straubhaar. 2006. "Symmetric time warping, Boltzmann pair probabilities and functional genomics." *Journal of Mathematical Biology* 53, no. 1 (July): 135–161.

Colby, Bruce N. 1992. "Spectral deconvolution for overlapping GC/MS components." *Journal of the American Society for Mass Spectrometry* 3, no. 5 (July): 558–562.

Comisarow, Melvin B., and Alan G. Marshall. 1974. "Fourier transform ion cyclotron resonance spectroscopy." *Chemical Physics Letters* 25 (2): 282–283.

Côté, Richard G., Florian Reisinger, and Lennart Martens. 2010. "jmzML, an open-source Java API for mzML, the PSI standard for MS data." *PROTEOMICS* 10 (7): 1332–1335.

Den Dool, H. van, and P. Kratz. 1963. "A generalization of the retention index system including linear temperature programmed gas-liquid partition chromatography." *Journal of Chromatography A* 11:463–471.

Dettmer, Katja, Pavel A. Aronov, and Bruce D. Hammock. 2007. "Mass spectrometry-based metabolomics." *Mass Spectrometry Reviews* 26 (1): 51–78.

Deutsch, Eric. 2008. "mzML: A single, unifying data format for mass spectrometer output." *PROTEOMICS* 8 (14): 2776–2777.

Deutsch, Eric W. 2012. "File Formats Commonly Used in Mass Spectrometry Proteomics." *Molecular & Cellular Proteomics* 11, no. 12 (December): 1612–1621.

Deutsch, Eric W., Luis Mendoza, David Shteynberg, Terry Farrah, Henry Lam, Natalie Tasman, Zhi Sun, Erik Nilsson, Brian Pratt, Bryan Prazen, Jimmy K. Eng, Daniel B. Martin, Alexey I. Nesvizhskii, and Ruedi Aebersold. 2010. "A guided tour of the Trans-Proteomic Pipeline." *PROTEOMICS* 10 (6): 1150–1159.

Doebbe, Anja, Matthias Keck, Marco La Russa, Jan H. Mussgnug, Ben Hankamer, Ercan Tekçe, Karsten Niehaus, and Olaf Kruse. 2010. "The Interplay of Proton, Electron, and Metabolite Supply for Photosynthetic H_2 Production in *Chlamydomonas reinhardtii*." *Journal of Biological Chemistry* 285, no. 39 (September): 30247–30260.

Du, Nan, Bin Wu, Liutong Xu, Bai Wang, and Xin Pei. 2006. "A Parallel Algorithm for Enumerating All Maximal Cliques in Complex Network." In *Sixth IEEE International Conference on Data Mining Workshops, 2006. ICDM Workshops 2006*, 320–324.

Du, Pan, Warren A. Kibbe, and Simon M. Lin. 2006. "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching." *Bioinformatics* 22, no. 17 (September): 2059–2065.

Dunn, Warwick B., and David. I. Ellis. 2005. "Metabolomics: Current analytical platforms and methodologies." *TrAC Trends in Analytical Chemistry* 24, no. 4 (April): 285–294.

Eilers, Paul H. C. 2004. "Parametric Time Warping." *Analytical Chemistry* 76, no. 2 (January): 404–411.

Eppstein, David, Michael T. Goodrich, and Jonathan Z. Sun. 2005. "The Skip Quadtree: A Simple Dynamic Data Structure for Multidimensional Data." In *Proceedings of the Twenty-first Annual Symposium on Computational Geometry*, 296–305. SCG '05. New York, NY, USA: ACM.

Erickson, Britt. 2000. "Government and Society: ANDI MS standard finalized." *Analytical Chemistry* 72, no. 3 (February): pages.

Fiehn, Oliver. 2002. "Metabolomics–the link between genotypes and phenotypes." *Plant molecular biology* 48, nos. 1-2 (January): 155–171.

Finkel, Raphael A., and Jon L. Bentley. 1974. "Quad trees a data structure for retrieval on composite keys." *Acta Informatica* 4, no. 1 (March): 1–9.

Fischer, Bernd, Volker Roth, and Joachim Buhmann. 2007. "Time-series alignment by non-negative multiple generalized canonical correlation analysis." *BMC Bioinformatics* 8, no. Suppl 10 (December): S4.

Fleischmann, R. D., M. D. Adams, O. White, R. A. Clayton, E. F. Kirkness, A. R. Kerlavage, C. J. Bult, J. F. Tomb, B. A. Dougherty, J. M. Merrick, and Et Al. 1995. "Whole-genome random sequencing and assembly of Haemophilus influenzae Rd." *Science* 269, no. 5223 (July): 496–512.

Foster, Ian T. 2005. "Globus Toolkit Version 4: Software for Service-Oriented Systems." In *IFIP International Conference on Network and Parallel Computing,* edited by Hai Jin, Daniel A. Reed, and Wenbin Jiang, 2–13. Berlin, Heidelberg: Springer.

Fredriksson, Mattias J., Patrik Petersson, Bengt-Olof Axelsson, and Dan Bylund. 2009. "An automatic peak finding method for LC-MS data using Gaussian second derivative filtering." *Journal of Separation Science* 32 (22): 3906–3918.

Geiser, Fritz. 2004. "Metabolic rate and body temperature reduction during hibernation and daily torpor." PMID: 14977403, *Annual review of physiology* 66:239–274.

Goodacre, Royston. 2005. "Metabolomics – the way forward." *Metabolomics* 1, no. 1 (March): 1–2.

Gosling, James. 2013. *The Java language specification.* Harlow: Prentice Hall.

Greef, Jan van der, Thomas Hankemeier, and Robert N. McBurney. 2006. "Metabolomics-based systems biology and personalized medicine: moving towards n = 1 clinical trials?" *Pharmacogenomics* 7, no. 7 (October): 1087–1094.

Griffiths, Jennifer. 2008. "A Brief History of Mass Spectrometry." *Analytical Chemistry* 80, no. 15 (August): 5678–5683.

Gross, Jürgen H. 2011. *Mass Spectrometry - A Textbook.* 2nd ed. Berlin, Heidelberg: Springer.

Halket, John M., Anna Przyborowska, Stephen E. Stein, W. Gary Mallard, Stephen Down, and Ronald A. Chalmers. 1999. "Deconvolution gas chromatography/-mass spectrometry of urinary organic acids – potential for pattern recognition and automated identification of metabolic disorders." *Rapid Communications in Mass Spectrometry* 13 (4): 279–284.

Harrigan, George G., and Royston Goodacre, eds. 2003. *Metabolic Profiling: Its Role in Biomarker Discovery and Gene Function Analysis.* Dordrecht: Kluwer, January.

Haug, Kenneth, Reza M. Salek, Pablo Conesa, Janna Hastings, Paula de Matos, Mark Rijnbeek, Tejasvi Mahendraker, Mark Williams, Steffen Neumann, Philippe Rocca-Serra, Eamonn Maguire, Alejandra González-Beltrán, Susanna-Assunta Sansone, Julian L. Griffin, and Christoph Steinbeck. 2013. "MetaboLights—an open-access general-purpose repository for metabolomics studies and associated meta-data." *Nucleic Acids Research* 41, no. D1 (January): D781–D786.

Hauschild, Anne-Christin, Dominik Kopczynski, Marianna D'Addario, Jörg Ingo Baumbach, Sven Rahmann, and Jan Baumbach. 2013. "Peak Detection Method Evaluation for Ion Mobility Spectrometry by Using Machine Learning Approaches." *Metabolites* 3, no. 2 (April): 277–293.

Hemschemeier, Anja, Swanny Fouchard, Laurent Cournac, Gilles Peltier, and Thomas Happe. 2008. "Hydrogen production by *Chlamydomonas reinhardtii*: an elaborate interplay of electron sources and sinks." *Planta* 227, no. 2 (January): 397–407.

Hoffmann, Nils, Matthias Keck, Heiko Neuweger, Mathias Wilhelm, Petra Högy, Karsten Niehaus, and Jens Stoye. 2012. "Combining peak- and chromatogram-based retention time alignment algorithms for multiple chromatography-mass spectrometry datasets." *BMC Bioinformatics* 13 (August): 214.

Hoffmann, Nils, and Jens Stoye. 2009. "ChromA: signal-based retention time alignment for chromatography–mass spectrometry data." *Bioinformatics* 25, no. 16 (August): 2080–2081.

———. 2012. "Generic Software Frameworks for GC-MS Based Metabolomics." In *Metabolomics,* edited by Ute Roessner, 73–98. Rijeka: InTech, February.

Hoffmann, Nils, Mathias Wilhelm, Anja Doebbe, Karsten Niehaus, and Jens Stoye. 2014. "BiPACE 2D—graph-based multiple alignment for comprehensive 2D gas chromatography-mass spectrometry." *Bioinformatics* 30, no. 7 (April): 988–995.

Högy, Petra, Matthias Keck, Karsten Niehaus, Jürgen Franzaring, and Andreas Fangmeier. 2010. "Effects of atmospheric CO_2 enrichment on biomass, yield and low molecular weight metabolites in wheat grain." *Journal of Cereal Science* 52, no. 2 (September): 215–220.

Högy, Petra, Herbert Wieser, Peter Köhler, Klaus Schwadorf, Jörn Breuer, Jürgen Franzaring, Russell B. Muntifering, and Andreas Fangmeier. 2009. "Effects of elevated CO_2 on grain yield and quality of wheat: results from a 3-year free-air CO_2 enrichment experiment." *Plant Biology* 11:60–69.

Hu, Qizhi, Robert J. Noll, Hongyan Li, Alexander Makarov, Mark Hardman, and R. Graham Cooks. 2005. "The Orbitrap: a new mass spectrometer." *Journal of Mass Spectrometry* 40 (4): 430–443.

Hufsky, Franziska, Martin Rempt, Florian Rasche, Georg Pohnert, and Sebastian Böcker. 2012. "De novo analysis of electron impact mass spectra using fragmentation trees." *Analytica Chimica Acta* 739 (August): 67–76.

Hull, Duncan, Katy Wolstencroft, Robert Stevens, Carole Goble, Mathew R. Pocock, Peter Li, and Tom Oinn. 2006. "Taverna: a tool for building and running workflows of services." *Nucleic Acids Research* 34, no. Web Server (July): W729–W732.

Hummel, Jan, Joachim Selbig, Dirk Walther, and Joachim Kopka. 2007. "The Golm Metabolome Database: a database for GC-MS based metabolite profiling." In *Metabolomics,* edited by Jens Nielsen and Michael C. Jewett, 75–95. Topics in Current Genetics 18. Berlin, Heidelberg: Springer, January.

Hummel, Jan, Nadine Strehmel, Joachim Selbig, Dirk Walther, and Joachim Kopka. 2010. "Decision tree supported substructure prediction of metabolites from GC-MS profiles." *Metabolomics* 6, no. 2 (June): 322–333.

Itakura, Fumitada. 1975. "Minimum prediction residual principle applied to speech recognition." *IEEE Transactions on Acoustics, Speech and Signal Processing* 23, no. 1 (February): 67–72.

Jaitly, Navdeep, Matthew E. Monroe, Vladislav A. Petyuk, Therese R. W. Clauss, Joshua N. Adkins, and Richard D. Smith. 2006. "Robust Algorithm for Alignment of Liquid Chromatography-Mass Spectrometry Analyses in an Accurate Mass and Time Tag Data Analysis Pipeline." *Analytical Chemistry* 78, no. 21 (November): 7397–7409.

Jeong, Jaesik, Xue Shi, Xiang Zhang, Seongho Kim, and Changyu Shen. 2012. "Model-based peak alignment of metabolomic profiling from comprehensive two-dimensional gas chromatography mass spectrometry." *BMC Bioinformatics* 13, no. 1 (February): 27.

Johnson, Kevin J., Bryan J. Prazen, Donald C. Young, and Robert E. Synovec. 2004. "Quantification of naphthalenes in jet fuel with GC×GC/Tri-PLS and windowed rank minimization retention time alignment." *Journal of Separation Science* 27 (5-6): 410–416.

Jonsson, Pär, Annika I. Johansson, Jonas Gullberg, Johan Trygg, Jiye A, Bjørn Grung, Stefan Marklund, Michael Sjöström, Henrik Antti, and Thomas Moritz. 2005. "High-Throughput Data Analysis for Detecting and Identifying Differences between Samples in GC/MS-Based Metabolomic Analyses." *Analytical Chemistry* 77, no. 17 (September): 5635–5642.

Kallio, Minna, Maarit Kivilompolo, Sami Varjo, Matti Jussila, and Tuulia Hyötyläinen. 2009. "Data analysis programs for comprehensive two-dimensional chromatography." *Journal of Chromatography A* 1216, no. 14 (April): 2923–2927.

Kanehisa, Minoru, Susumu Goto, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. 2013. "Data, information, knowledge and principle: back to metabolism in KEGG." *Nucleic Acids Research* (November): gkt1076.

Kankainen, Matti, Peddinti Gopalacharyulu, Liisa Holm, and Matej Orešič. 2011. "MPEA—metabolite pathway enrichment analysis." *Bioinformatics* 27, no. 13 (July): 1878–1879.

Karp, Richard M. 1972. "Reducibility Among Combinatorial Problems." In *Complexity of Computer Computations,* edited by R. E. Miller and J. W. Thatcher, 85–103. New York, London: Plenum Press.

Kastenmüller, Gabi, Werner Römisch-Margl, Brigitte Wägele, Elisabeth Altmaier, and Karsten Suhre. 2011. "metaP-Server: A Web-Based Metabolomics Data Analysis Tool." *Journal of Biomedicine and Biotechnology* 2011:1–8.

Kernighan, Brian W., and Dennis M. Ritchie. 1988. *The C programming language / ANSI C Version.* Englewood Cliffs, N.J.: Prentice Hall.

Kessler, Nikolas, Heiko Neuweger, Anja Bonte, Georg Langenkämper, Karsten Niehaus, Tim W. Nattkemper, and Alexander Goesmann. 2013. "MeltDB 2.0–advances of the metabolomics software system." *Bioinformatics* 29, no. 19 (October): 2452–2459.

Kessner, Darren, Matt Chambers, Robert Burke, David Agus, and Parag Mallick. 2008. "ProteoWizard: open source software for rapid proteomics tools development." *Bioinformatics* 24, no. 21 (November): 2534–2536.

Kim, Seongho, Aiqin Fang, Bing Wang, Jaesik Jeong, and Xiang Zhang. 2011. "An optimal peak alignment for comprehensive two-dimensional gas chromatography mass spectrometry using mixture similarity measure." *Bioinformatics* 27, no. 12 (June): 1660–1666.

Kim, Seongho, Imhoi Koo, Aiqin Fang, and Xiang Zhang. 2011. "Smith-Waterman peak alignment for comprehensive two-dimensional gas chromatography-mass spectrometry." *BMC Bioinformatics* 12 (June): 235.

Kim, Seongho, and Xiang Zhang. 2013. "Comparative Analysis of Mass Spectral Similarity Measures on Peak Alignment for Comprehensive Two-Dimensional Gas Chromatography Mass Spectrometry." *Computational and Mathematical Methods in Medicine* 2013 (September).

Kitteringham, Neil R., Rosalind E. Jenkins, Catherine S. Lane, Victoria L. Elliott, and B. Kevin Park. 2009. "Multiple reaction monitoring for quantitative biomarker analysis in proteomics and metabolomics." *Journal of Chromatography B* 877 (13): 1229–1239.

Koal, T., and H.-P. Deigner. 2010. "Challenges in Mass Spectrometry Based Targeted Metabolomics." *Current Molecular Medicine* 10, no. 2 (March): 216–226.

Koek, Maud M., Frans M. van der Kloet, Robert Kleemann, Teake Kooistra, Elwin R. Verheij, and Thomas Hankemeier. 2011. "Semi-automated non-target processing in GC × GC–MS metabolomics analysis: applicability for biomedical studies." *Metabolomics* 7, no. 1 (March): 1–14.

Koek, Maud M., Bas Muilwijk, Mariët J. van der Werf, and Thomas Hankemeier. 2006. "Microbial Metabolomics with Gas Chromatography/Mass Spectrometry." *Analytical Chemistry* 78, no. 4 (February): 1272–1281.

Kondrat, Richard W., Gary A. McClusky, and R. Graham Cooks. 1978. "Multiple reaction monitoring in mass spectrometry/mass spectrometry for direct analysis of complex mixtures." *Analytical Chemistry* 50 (14): 2017–2021.

Krebs, Melissa D., Robert D. Tingley, Julie E. Zeskind, Maria E. Holmboe, Joung-Mo Kang, and Cristina E. Davis. 2006. "Alignment of gas chromatography-mass spectrometry data by landmark selection from complex chemical mixtures." *Chemometrics and Intelligent Laboratory Systems* 81 (1): 74–81.

Kruskal, Joseph B., and Mark Liberman. 1983. *The symmetric time-warping problem: from continuous to discrete.* Edited by D. Sankoff and J. Kruskal. Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Stanford: CSLI Publications.

Kueh, A.J., Philip J. Marriott, Paul M. Wynne, and John H. Vine. 2003. "Application of comprehensive two-dimensional gas chromatography to drugs analysis in doping control." *Journal of Chromatography A* 1000, no. 1–2 (June): 109–124.

Lahl, Uwe, and Katrin Anne Hawxwell. 2006. "REACH—The New European Chemicals Law." *Environmental Science & Technology* 40 (23): 7115–7121.

Lange, Eva, Clemens Gröpl, Ole Schulz-Trieglaff, Andreas Leinenbach, Christian Huber, and Knut Reinert. 2007. "A geometric approach for the alignment of liquid chromatography—mass spectrometry data." *Bioinformatics* 23, no. 13 (July): i273–i281.

Lange, Eva, Ralf Tautenhahn, Steffen Neumann, and Clemens Gröpl. 2008. "Critical assessment of alignment procedures for LC-MS proteomics and metabolomics measurements." *BMC Bioinformatics* 9, no. 1 (September): 375.

Latha, Indu, Stephen E. Reichenbach, and Qingping Tao. 2011. "Comparative analysis of peak-detection techniques for comprehensive two-dimensional chromatography." *Journal of Chromatography A* 1218, no. 38 (September): 6792–6798.

Likić, Vladimir. 2009. "Extraction of pure components from overlapped signals in gas chromatography-mass spectrometry (GC-MS)." *BioData Mining* 2, no. 1 (October): 6.

Linke, Burkhard, Robert Giegerich, and Alexander Goesmann. 2011. "Conveyor: a workflow engine for bioinformatic analyses." *Bioinformatics* 27, no. 7 (April): 903–911.

Listgarten, Jennifer, Radford M. Neal, Sam Roweis, and Andrew Emili. 2005. "Multiple alignment of continuous time series." In *Advances in Neural Information Processing Systems*, 17:817–824. Cambridge, MA: MIT Press, January.

Lockhart, David J., Helin Dong, Michael C. Byrne, Maximillian T. Follettie, Michael V. Gallo, Mark S. Chee, Michael Mittmann, Chunwei Wang, Michiko Kobayashi, Heidi Norton, and Eugene L. Brown. 1996. "Expression monitoring by hybridization to high-density oligonucleotide arrays." *Nature Biotechnology* 14 (13): 1675–1680.

Lommen, Arjen. 2009. "MetAlign: Interface-Driven, Versatile Metabolomics Tool for Hyphenated Full-Scan Mass Spectrometry Data Preprocessing." *Analytical Chemistry* 81, no. 8 (April): 3079–3086.

Lovelock, James. 2004. "Archer John Porter Martin CBE. 1 March 1910 — 28 July 2002 Elected F.R.S. 1950." *Biographical Memoirs of Fellows of the Royal Society* 50 (December): 157–170.

Martens, Lennart, Matthew Chambers, Marc Sturm, Darren Kessner, Fredrik Levander, Jim Shofstahl, Wilfred H. Tang, Andreas Römpp, Steffen Neumann, Angel D. Pizarro, Luisa Montecchi-Palazzi, Natalie Tasman, Mike Coleman, Florian Reisinger, Puneet Souda, Henning Hermjakob, Pierre-Alain Binz, and Eric W. Deutsch. 2011. "mzML—a Community Standard for Mass Spectrometry Data." *Molecular & Cellular Proteomics* 10, no. 1 (January).

Matos, João T.V., Regina M.B.O. Duarte, and Armando C. Duarte. 2012. "Trends in data processing of comprehensive two-dimensional chromatography: State of the art." *Journal of Chromatography B* 910:31–45.

Matthew, Timmins, Wenxu Zhou, Jens Rupprecht, Lysha Lim, Skye R. Thomas-Hall, Anja Doebbe, Olaf Kruse, Ben Hankamer, Ute C. Marx, Steven M. Smith, and Peer M. Schenk. 2009. "The Metabolome of *Chlamydomonas reinhardtii* following Induction of Anaerobic H_2 Production by Sulfur Depletion." *Journal of Biological Chemistry* 284, no. 35 (August): 23415–23425.

Matthews, Lynn, and Todd Miller. 2000. "ASTM Protocols for Analytical Data Interchange." *Journal of the Association for Laboratory Automation* 5, no. 5 (October): 60–61.

McLafferty, Fred W. 1959. "Mass Spectrometric Analysis. Molecular Rearrangements." *Analytical Chemistry* 31, no. 1 (January): 82–87.

McWilliam, I. G., and R. A. Dewar. 1958. "Flame Ionization Detector for Gas Chromatography." *Nature* 181, no. 4611 (March): 760–760.

Meléndez-Hevia, Enrique, Thomas G. Waddell, and Marta Cascante. 1996. "The puzzle of the Krebs citric acid cycle: Assembling the pieces of chemically feasible reactions, and opportunism in the design of metabolic pathways during evolution." *Journal of Molecular Evolution* 43, no. 3 (September): 293–303.

Melis, Anastasios, Liping Zhang, Marc Forestier, Maria L. Ghirardi, and Michael Seibert. 2000. "Sustained Photobiological Hydrogen Gas Production upon Reversible Inactivation of Oxygen Evolution in the Green AlgaChlamydomonas reinhardtii." *Plant Physiology* 122, no. 1 (January): 127–136.

Mesarović, Mihajlo D. 1968. *Systems Theory and Biology–Proceedings of the III Systems Symposium at Case Institute of Technology.* Edited by Mihajlo D. Mesarović. Berlin Heidelberg: Springer, January.

Miura, Daisuke, Yukiko Tsuji, Katsutoshi Takahashi, Hiroyuki Wariishi, and Kazunori Saito. 2010. "A Strategy for the Determination of the Elemental Composition by Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Based on Isotopic Peak Ratios." *Analytical Chemistry* 82, no. 13 (July): 5887–5891.

Moayyeri, Alireza, Christopher J. Hammond, Ana M. Valdes, and Timothy D. Spector. 2013. "Cohort Profile: TwinsUK and Healthy Ageing Twin Study." PMID: 22253318, *International Journal of Epidemiology* 42, no. 1 (February): 76–85.

Moeller, M. R., P. Fey, and R. Wennig. 1993. "Simultaneous determination of drugs of abuse (opiates, cocaine and amphetamine) in human hair by GCMS and its application to a methadone treatment program." *Forensic Science International* 63 (1–3): 185–206.

Mondello, Luigi, Peter Quinto Tranchida, Paola Dugo, and Giovanni Dugo. 2008. "Comprehensive two-dimensional gas chromatography-mass spectrometry: A review." *Mass Spectrometry Reviews* 27 (2): 101–124.

Morhardt, Emil J. 1970. "Body temperatures of white-footed mice (Peromyscus sp.) during daily torpor." *Comparative Biochemistry and Physiology* 33 (2): 423–439.

Muñoz, Arrate, Raphaël Ertlé, and Michael Unser. 2002. "Continuous wavelet transform with arbitrary scales and O(N) complexity." *Signal Processing* 82 (5): 749–757.

Neumann, Steffen, and Sebastian Böcker. 2010. "Computational mass spectrometry for metabolomics: Identification of metabolites and small molecules." *Analytical and Bioanalytical Chemistry* 398, nos. 7-8 (December): 2779–2788.

Neuweger, Heiko, Stefan P. Albaum, Michael Dondrup, Marcus Persicke, Tony Watt, Karsten Niehaus, Jens Stoye, and Alexander Goesmann. 2008. "MeltDB: a software platform for the analysis and integration of metabolomics experiment data." *Bioinformatics* 24, no. 23 (December): 2726–2732.

Neuweger, Heiko, Marcus Persicke, Stefan Albaum, Thomas Bekel, Michael Dondrup, Andrea Hüser, Jörn Winnebald, Jessica Schneider, Jörn Kalinowski, and Alexander Goesmann. 2009. "Visualizing post genomics data-sets on customized pathway maps by ProMeTra – aeration-dependent gene expression and metabolism of Corynebacterium glutamicum as an example." *BMC Systems Biology* 3, no. 1 (August): 82.

Nielsen, Jens Høiriis, and Michael C. Jewett, eds. 2007. *Metabolomics: A Powerful Tool in Systems Biology*. Topics in Current Genetics 18. Berlin, Heidelberg: Springer, September.

Oh, Cheolhwan, Xiaodong Huang, Fred E Regnier, Charles Buck, and Xiang Zhang. 2008. "Comprehensive two-dimensional gas chromatography/time-of-flight mass spectrometry peak sorting algorithm." *Journal of Chromatography A* 1179, no. 2 (February): 205–215.

Oliver, Stephen G., Michael K. Winson, Douglas B. Kell, and Frank Baganz. 1998. "Systematic functional analysis of the yeast genome." *Trends in Biotechnology* 16, no. 9 (September): 373–378.

Orchard, Sandra, Henning Hermjakob, Randall K. Julian, Kai Runte, David Sherman, Jérôme Wojcik, Weimin Zhu, and Rolf Apweiler. 2004. "Common interchange standards for proteomics data: Public availability of tools and schema. Report on the Proteomic Standards Initiative Workshop, 2nd Annual HUPO Congress, Montreal, Canada, 8–11th October 2003." *PROTEOMICS* 4 (2): 490–491.

Orchard, Sandra, Henning Hermjakob, Chris F. Taylor, Frank Potthast, Phil Jones, Weimin Zhu, Randall K. Julian Jr, and Rolf Apweiler. 2005. "Second Proteomics Standards Initiative Spring Workshop." *Expert Review of Proteomics* 2, no. 3 (June): 287–289.

Overbeek, Ross, Michael Fonstein, Mark D'Souza, Gordon D. Pusch, and Natalia Maltsev. 1999. "The use of gene clusters to infer functional coupling." *Proceedings of the National Academy of Sciences of the U.S.A.* 96, no. 6 (March): 2896–2901.

Pedrioli, Patrick G. A., Jimmy K. Eng, Robert Hubley, Mathijs Vogelzang, Eric W. Deutsch, Brian Raught, Brian Pratt, Erik Nilsson, Ruth H. Angeletti, Rolf Apweiler, Kei Cheung, Catherine E. Costello, Henning Hermjakob, Sequin Huang, Randall K. Julian, Eugene Kapp, Mark E. McComb, Stephen G. Oliver, Gilbert Omenn, Norman W. Paton, Richard Simpson, Richard Smith, Chris F. Taylor, Weimin Zhu, and Ruedi Aebersold. 2004. "A common open representation of mass spectrometry data and its application to proteomics research." *Nature Biotechnology* 22 (11): 1459–1466.

Percival, Donald B., and Andrew T. Walden. 2000. *Wavelet Methods for Time Series Analysis*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press.

Peters, Sonja, Gabriel Vivó-Truyols, Philip J. Marriott, and Peter J. Schoenmakers. 2007. "Development of an algorithm for peak detection in comprehensive two-dimensional chromatography." *Journal of Chromatography A* 1156, no. 1–2 (July): 14–24.

Pierce, Karisa M., Jamin C. Hoggard, Janiece L. Hope, Petrie M. Rainey, Andrew N. Hoofnagle, Rhona M. Jack, Bob W. Wright, and Robert E. Synovec. 2006. "Fisher Ratio Method Applied to Third-Order Separation Data To Identify Significant Chemical Components of Metabolite Extracts." *Analytical Chemistry* 78, no. 14 (July): 5068–5075.

Pierce, Karisa M., Janiece L. Hope, Kevin J. Johnson, Bob W. Wright, and Robert E. Synovec. 2005. "Classification of gasoline data obtained by gas chromatography using a piecewise alignment algorithm combined with feature selection and principal component analysis." *Journal of Chromatography A* 1096, no. 1–2 (November): 101–110.

Pierce, Karisa M., Bob W. Wright, and Robert E. Synovec. 2007. "Unsupervised parameter optimization for automated retention time alignment of severely shifted gas chromatographic data using the piecewise alignment algorithm." *Journal of Chromatography A* 1141, no. 1 (February): 106–116.

Pluskal, Tomáš, Sandra Castillo, Alejandro Villar-Briones, and Matej Orešič. 2010. "MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data." *BMC Bioinformatics* 11, no. 1 (July): 395.

Podwojski, Katharina, Arno Fritsch, Daniel C. Chamrad, Wolfgang Paul, Barbara Sitek, Kai Stühler, Petra Mutzel, Christian Stephan, Helmut E. Meyer, Wolfgang Urfer, Katja Ickstadt, and Jörg Rahnenführer. 2009. "Retention time alignment algorithms for LC/MS data must consider non-linear shifts." *Bioinformatics* 25, no. 6 (March): 758–764.

Prakash, Amol, Parag Mallick, Jeffrey Whiteaker, Heidi Zhang, Amanda Paulovich, Mark Flory, Hookeun Lee, Ruedi Aebersold, and Benno Schwikowski. 2006. "Signal Maps for Mass Spectrometry-based Comparative Proteomics." *Molecular & Cellular Proteomics* 5, no. 3 (March): 423–432.

Prince, John T., and Edward M. Marcotte. 2006. "Chromatographic Alignment of ESI-LC-MS Proteomics Data Sets by Ordered Bijective Interpolated Warping." *Analytical Chemistry* 78, no. 17 (September): 6140–6152.

Ramaker, Henk-Jan, Eric N.M. van Sprang, Johan A. Westerhuis, and Age K. Smilde. 2003. "Dynamic time warping of spectroscopic BATCH data." *Analytica Chimica Acta* 498, no. 1–2 (November): 133–153.

Reddy, Christopher M., J. Samuel Arey, Jeffrey S. Seewald, Sean P. Sylva, Karin L. Lemkau, Robert K. Nelson, Catherine A. Carmichael, Cameron P. McIntyre, Judith Fenwick, G. Todd Ventura, Benjamin A. S. Van Mooy, and Richard Camilli. 2011. "Composition and fate of gas and oil released to the water column during the Deepwater Horizon oil spill." *Proceedings of the National Academy of Sciences* (July).

Reichenbach, Stephen E., Xue Tian, Akwasi A. Boateng, Charles A. Mullen, Chiara Cordero, and Qingping Tao. 2013. "Reliable Peak Selection for Multisample Analysis with Comprehensive Two-Dimensional Chromatography." *Analytical Chemistry* 85 (10): 4974–4981.

Reichenbach, Stephen E., Xue Tian, Chiara Cordero, and Qingping Tao. 2012. "Features for non-targeted cross-sample analysis with comprehensive two-dimensional chromatography." *Journal of Chromatography A* 1226 (February): 140–148.

Reiner, E., L. E. Abbey, T. F. Moran, P. Papamichalis, and R. W. Schafer. 1979. "Characterization of normal human cells by pyrolysis gas chromatography mass spectrometry." *Biological Mass Spectrometry* 6 (11): 491–498.

Rew, Russ K., and Glenn P. Davis. 1990. "NetCDF: An Interface for Scientific Data Access." *IEEE Comput. Graph. Appl. Mag.* 10, no. 4 (July): 76–82.

Robinson, Mark, David De Souza, Woon Keen, Eleanor Saunders, Malcolm McConville, Terence Speed, and Vladimir Likić. 2007. "A dynamic programming approach for the alignment of signal peaks in multiple gas chromatography-mass spectrometry experiments." *BMC Bioinformatics* 8 (October): 419.

Rosgen, Bill, and Lorna Stewart. 2007. "Complexity Results on Graphs with Few Cliques." *Discrete Mathematics & Theoretical Computer Science* 9 (August): 127–136.

Sakoe, Hiroaki, and Seibi Chiba. 1978. "Dynamic programming algorithm optimization for spoken word recognition." *IEEE Transactions on Acoustics, Speech and Signal Processing* 26, no. 1 (February): 43–49.

Samet, Hanan. 2006. *Foundations of Multidimensional and Metric Data Structures.* Burlington: Morgan Kaufmann.

Savitzky, Abraham, and Marcel J. E. Golay. 1964. "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." *Analytical Chemistry* 36, no. 8 (July): 1627–1639.

Schatschneider, Sarah, Marcus Persicke, Steven Alexander Watt, Gerd Hublik, Alfred Pühler, Karsten Niehaus, and Frank-Jörg Vorhölter. 2013. "Establishment, in silico analysis, and experimental verification of a large-scale metabolic network of the xanthan producing Xanthomonas campestris pv. campestris strain B100." *Journal of Biotechnology* 167, no. 2 (August): 123–134.

Scheltema, Richard A., Andris Jankevics, Ritsert C. Jansen, Morris A. Swertz, and Rainer Breitling. 2011. "PeakML/mzMatch: A File Format, Java Library, R Library, and Tool-Chain for Mass Spectrometry Data Analysis." *Analytical chemistry* 83, no. 7 (April): 2786–2793.

Schenk, Peer M., Skye R. Thomas-Hall, Evan Stephens, Ute C. Marx, Jan H. Mussgnug, Clemens Posten, Olaf Kruse, and Ben Hankamer. 2008. "Second Generation Biofuels: High-Efficiency Microalgae for Biodiesel Production." *BioEnergy Research* 1, no. 1 (March): 20–43.

Schmidt, Matthew C., Nagiza F. Samatova, Kevin Thomas, and Byung-Hoon Park. 2009. "A scalable, parallel algorithm for maximal clique enumeration." *Journal of Parallel and Distributed Computing* 69, no. 4 (April): 417–428.

Shannon, Claude E. 1949. "Communication In The Presence Of Noise." *Proceedings of the Institute of Radio Engineers* 37 (1): 10–21.

Shellie, Robert A., Werner Welthagen, Jitka Zrostliková, Joachim Spranger, Michael Ristow, Oliver Fiehn, and Ralf Zimmermann. 2005. "Statistical methods for comparing comprehensive two-dimensional gas chromatography–time-of-flight mass spectrometry results: Metabolomic analysis of mouse tissue extracts." *Journal of Chromatography A* 1086, no. 1–2 (September): 83–90.

Shevchenko, Andrej, Ole N. Jensen, Alexandre V. Podtelejnikov, Francis Sagliocco, Matthias Wilm, Ole Vorm, Peter Mortensen, Anna Shevchenko, Helian Boucherie, and Matthias Mann. 1996. "Linking genome and proteome by mass spectrometry: Large-scale identification of yeast proteins from two dimensional gels." *Proceedings of the National Academy of Sciences* 93, no. 25 (December): 14440–14445.

Smith, Colin A., Elizabeth J. Want, Grace O'Maille, Ruben Abagyan, and Gary Siuzdak. 2006. "XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification." *Analytical Chemistry* 78, no. 3 (February): 779–787.

Stein, Stephen E. 1999. "An integrated method for spectrum extraction and compound identification from gas chromatography/mass spectrometry data." *Journal of the American Society for Mass Spectrometry* 10, no. 8 (August): 770–781.

Stein, Stephen E., and Donald R. Scott. 1994. "Optimization and testing of mass spectral library search algorithms for compound identification." *Journal of the American Society for Mass Spectrometry* 5, no. 9 (September): 859–866.

Strehmel, Nadine, Jan Hummel, Alexander Erban, Katrin Strassburg, and Joachim Kopka. 2008. "Retention index thresholds for compound matching in GC–MS metabolite profiling." *Journal of Chromatography B* 871, no. 2 (August): 182–190.

Stroustrup, Bjarne. 2013. *The C++ programming language.* Amsterdam: Addison-Wesley.

Sturm, Marc, Andreas Bertsch, Clemens Gröpl, Andreas Hildebrandt, Rene Hussong, Eva Lange, Nico Pfeifer, Ole Schulz-Trieglaff, Alexandra Zerck, Knut Reinert, and Oliver Kohlbacher. 2008. "OpenMS – An open-source software framework for mass spectrometry." *BMC Bioinformatics* 9, no. 1 (March): 163.

Styczynski, Mark P., Joel F. Moxley, Lily V. Tong, Jason L. Walther, Kyle L. Jensen, and Gregory N. Stephanopoulos. 2007. "Systematic Identification of Conserved Metabolites in GC/MS Data for Metabolomics and Biomarker Discovery." *Analytical Chemistry* 79, no. 3 (February): 966–973.

Sumner, Lloyd W., Pedro Mendes, and Richard A. Dixon. 2003. "Plant metabolomics: large-scale phytochemistry in the functional genomics era." *Phytochemistry* 62 (6): 817–836.

Sweldens, Wim. 1998. "The Lifting Scheme: A Construction of Second Generation Wavelets." *SIAM Journal on Mathematical Analysis* 29, no. 2 (March): 511–546.

Swoap, Steven J. 2008. "The pharmacology and molecular mechanisms underlying temperature regulation and torpor." *Biochemical Pharmacology* 76 (7): 817–824.

Tang-Liu, D. D., R. L. Williams, and S. Riegelman. 1983. "Disposition of caffeine and its metabolites in man." *Journal of Pharmacology and Experimental Therapeutics* 224, no. 1 (January): 180–185.

Tatusov, Roman L., Eugene V. Koonin, and David J. Lipman. 1997. "A Genomic Perspective on Protein Families." *Science* 278, no. 5338 (October): 631–637.

Tautenhahn, Ralf, Christoph Böttcher, and Steffen Neumann. 2007. "Annotation of LC/ESI-MS Mass Signals." In *Bioinformatics Research and Development*, edited by Sepp Hochreiter and Roland Wagner, 371–380. 10.1007/978-3-540-71233-6_29. Berlin, Heidelberg: Springer.

———. 2008. "Highly sensitive feature detection for high resolution LC/MS." *BMC Bioinformatics* 9, no. 1 (November): 504.

Tautenhahn, Ralf, Gary J. Patti, Duane Rinehart, and Gary Siuzdak. 2012. "XCMS Online: A Web-Based Platform to Process Untargeted Metabolomic Data." *Analytical Chemistry* 84, no. 11 (June): 5035–5039.

Tian, Jing, Chunyun Shi, Peng Gao, Kailong Yuan, Dawei Yang, Xin Lu, and Guowang Xu. 2008. "Phenotype differentiation of three E. coli strains by GC-FID and GC–MS based metabolomics." *Journal of Chromatography B* 871, no. 2 (August): 220–226.

Tohge, Takayuki, and Alisdair R. Fernie. 2009. "Web-based resources for mass-spectrometry-based metabolomics: A user's guide." *Phytochemistry* 70 (4): 450–456.

Ventura, G. Todd, Gregory J. Hall, Robert K. Nelson, Glenn S. Frysinger, Bhavani Raghuraman, Andrew E. Pomerantz, Oliver C. Mullins, and Christopher M. Reddy. 2011. "Analysis of petroleum compositional similarity using multiway principal components analysis (MPCA) with comprehensive two-dimensional gas chromatographic data." *Journal of Chromatography A* 1218 (18): 2584–2592.

Vestal, Marvin L. 2009. "Modern MALDI time-of-flight mass spectrometry." *Journal of Mass Spectrometry* 44 (3): 303–317.

Vial, Jérôme, Hicham Noçairi, Patrick Sassiat, Sreedhar Mallipatu, Guillaume Cognon, Didier Thiébaut, Béatrice Teillet, and Douglas N Rutledge. 2009. "Combination of dynamic time warping and multivariate analysis for the comparison of comprehensive two-dimensional gas chromatograms: application to plant extracts." *Journal of Chromatography A* 1216, no. 14 (April): 2866–2872.

Vivó-Truyols, Gabriel. 2012. "Bayesian Approach for Peak Detection in Two-Dimensional Chromatography." *Analytical Chemistry* 84 (6): 2622–2630.

Vivó-Truyols, Gabriel, and Hans-Gerd Janssen. 2010. "Probability of failure of the watershed algorithm for peak detection in comprehensive two-dimensional chromatography." *Journal of Chromatography A* 1217, no. 8 (February): 1375–1385.

Walker, James S. 1997. "Fourier analysis and wavelet analysis." *Notices of the AMS* 44 (6): 658–670.

Wang, San-Yuan, Tsung-Jung Ho, Ching-Hua Kuo, and Yufeng J. Tseng. 2010. "Chromaligner: a web server for chromatogram alignment." *Bioinformatics* 26, no. 18 (September): 2338–2339.

Wang, Yanli, Jewen Xiao, Tugba O. Suzek, Jian Zhang, Jiyao Wang, and Stephen H. Bryant. 2009. "PubChem: a public information system for analyzing bioactivities of small molecules." *Nucleic Acids Research* 37, no. suppl 2 (July): W623–W633.

Weckwerth, Wolfram, ed. 2007. *Metabolomics: Methods and Protocols.* Methods in Molecular Biology 358. Totowa: Humana Press.

———. 2011. "Unpredictability of metabolism—the key role of metabolomics science in combination with next-generation genome sequencing." *Analytical and Bioanalytical Chemistry* 400, no. 7 (June): 1967–1978.

Wei, Xiaoli, Xue Shi, Imhoi Koo, Seongho Kim, Robin H. Schmidt, Gavin E. Arteel, Walter H. Watson, Craig McClain, and Xiang Zhang. 2013. "MetPP: a computational platform for comprehensive two-dimensional gas chromatography time-of-flight mass spectrometry-based metabolomics." *Bioinformatics* 29, no. 14 (July): 1786–1792.

Wenig, Philip, and Juergen Odermatt. 2010. "OpenChrom: a cross-platform open source software for the mass spectrometric analysis of chromatographic data." *BMC Bioinformatics* 11, no. 1 (July): 405.

Weston, Andrea D., and Leroy Hood. 2004. "Systems Biology, Proteomics, and the Future of Health Care: Toward Predictive, Preventative, and Personalized Medicine." *Journal of Proteome Research* 3, no. 2 (April): 179–196.

Wiklund, Susanne, Erik Johansson, Lina Sjostrom, Ewa J. Mellerowicz, Ulf Edlund, John P. Shockcor, Johan Gottfries, Thomas Moritz, and Johan Trygg. 2008. "Visualization of GC/TOF-MS-Based Metabolomics Data for Identification of Biochemically Interesting Compounds Using OPLS Class Models." *Analytical Chemistry* 80, no. 1 (January): 115–122.

Wilhelm, Mathias, Marc Kirchner, Judith A. J. Steen, and Hanno Steen. 2011. "mz5: space- and time-efficient storage of mass spectrometry data sets." *Molecular & Cellular Proteomics* (September).

Windig, Willem, J. Martin Phalp, and Alan W. Payne. 1996. "A Noise and Background Reduction Method for Component Detection in Liquid Chromatography/Mass Spectrometry." *Analytical Chemistry* 68, no. 20 (January): 3602–3606.

Wishart, David S., Craig Knox, An Chi Guo, Roman Eisner, Nelson Young, Bijaya Gautam, David D. Hau, Nick Psychogios, Edison Dong, Souhaila Bouatra, Rupasri Mandal, Igor Sinelnikov, Jianguo Xia, Leslie Jia, Joseph A. Cruz, Emilia Lim, Constance A. Sobsey, Savita Shrivastava, Paul Huang, Philip Liu, Lydia Fang, Jun Peng, Ryan Fradette, Dean Cheng, Dan Tzur, Melisa Clements, Avalyn Lewis, Andrea De Souza, Azaret Zuniga, Margot Dawe, Yeping Xiong, Derrick Clive, Russ Greiner, Alsu Nazyrova, Rustem Shaykhutdinov, Liang Li, Hans J. Vogel, and Ian Forsythe. 2009. "HMDB: a knowledgebase for the human metabolome." PMID: 18953024, *Nucleic Acids Research* 37, no. suppl 1 (January): D603–D610.

Xia, Jianguo, Igor V. Sinelnikov, and David S. Wishart. 2011. "MetATT: a web-based metabolomics tool for analyzing time-series and two-factor datasets." *Bioinformatics* 27, no. 17 (September): 2455–2456.

Xia, Jianguo, and David S. Wishart. 2010. "MetPA: a web-based metabolomics tool for pathway analysis and visualization." *Bioinformatics* 26, no. 18 (September): 2342–2344.

Zhang, Xiang, John M. Asara, Jiri Adamec, Mourad Ouzzani, and Ahmed K. Elmagarmid. 2005. "Data pre-processing in liquid chromatography–mass spectrometry-based proteomics." *Bioinformatics* 21, no. 21 (January): 4054–4059.

Zhang, Zhi-Min, Yi-Zeng Liang, Hong-Mei Lu, Bin-Bin Tan, Xiao-Na Xu, and Miguel Ferro. 2012. "Multiscale peak alignment for chromatographic datasets." *Journal of Chromatography A* 1223 (February): 93–106.

Zwet, Michail S. 1906. "Physikalisch-Chemische Studien über das Chlorophyll. Die Adsorption." In *Berichte der Deutschen Botanischen Gesellschaft*, 24:316–326. Stuttgart: Gebrüder Borntræger.

Acronyms

BIPACE **BI**directional best hits **P**eak **A**ssignment and **C**luster **E**xtension. 29, 37, 40, 45, 47, 48, 51, 52, 55, 57–65, 67–70, 72–74, 123, 124, 133, 145, 146

BIPACE 2D BIPACE with two-dimensional retention time. 136, 146

CHROMA **Chrom**atogram **A**lignment. 30, 36, 175, 177

CHROMA4D **Chrom**atogram **A**lignment for **4D** GC×GC-MS data. 76, 78, 175, 180

CROSS **C**ommon **R**untime **O**bject **S**upport **S**ystem. 36, 111, 112, 114–117, 120, 124, 131, 147

CEMAPP-DTW **CE**nter-star **M**ultiple **A**lignment by **P**airwise **P**artitioned **D**ynamic **T**ime **W**arping. 37, 40, 54, 58–61, 63–67, 69–74, 121, 123, 133, 145

GC-FID gas chromatography-flame ionization detector. 34, 122, 147

GC×GC comprehensive two-dimensional gas chromatography. 13, 16, 27, 28, 80, 127

GC×GC-MS comprehensive two-dimensional gas chromatography-mass spectrometry. 2, 3, 14, 19, 21, 24, 25, 27, 28, 75, 76, 78–81, 85, 86, 88–94, 109, 114, 120, 121, 123, 124, 133, 134, 141, 145–148, 171, 175

GC-MS gas chromatography-mass spectrometry. xiii, 2, 3, 14, 18–20, 22, 23, 25–27, 29–34, 36, 37, 40, 41, 43, 44, 52, 53, 58, 65, 73, 75, 91, 93, 111, 120, 145, 147, 148, 171, 175, 176, 178, 184

GC gas chromatography. xiii, 2, 7, 11–13, 16, 18, 19, 22, 34, 75, 171, 183

LC liquid chromatography. 7, 11, 12, 16, 18, 20, 22, 121, 171, 173

LC×LC comprehensive two-dimensional liquid chromatography. 127

LC×LC-MS comprehensive two-dimensional liquid chromatography-mass spectrometry. 121, 147

LC-MS liquid chromatography-mass spectrometry. 18–20, 22, 23, 25–27, 29–31, 33, 36, 37, 39, 40, 52, 53, 58, 91, 121, 148, 171

MALTCMS **M**odular **A**pplication **T**oolkit for **C**hromatography-**M**ass Spectrometry. 29, 30, 35, 36, 111, 112, 114, 116, 117, 120–122, 124, 126, 128, 129, 131, 132, 136, 137, 142, 147–149, 172, 175, 177, 182, 184

MALTCMS AP MALTCMS for **A**nalytical **P**yrolysis. 182

MAUI **Ma**ltcms **U**ser **I**nterface. 129–134, 136, 142, 144, 147–149, 184

MPAXS **M**altcms **Pa**rallel E**x**ecution **S**ystem. 111, 117–120, 131, 147

AC alternating current. 17

ANOVA analysis of variance. 28, 33, 76, 114, 134, 136, 143

APCI atmospheric pressure chemical ionization. 12

API application programming interface. 112, 116, 120, 131, 132, 142

ASTM American Society for Testing and Materials. 23, 124

BBH bidirectional best-hit. 36, 46, 48–52, 146

CE capillary electrophoresis. 19

CI chemical ionization. 16, 17

CODA component detection algorithm. 34

COW correlation optimized warping. 26, 39

CPU central processing unit. 112

CSV comma separated value. 36, 37, 78, 122, 133, 134

CV controlled vocabulary. 126

CWT continuous wavelet transform. viii, 80–86, 88, 89, 121, 122, 147

DC direct current. 17

DRMAA distributed resource management application api. 117, 118

DTW dynamic time warping. 26, 33, 36, 39, 40, 52–58, 73, 111, 123, 145, 177, 178, 181, 194, 195, 200, 202

EI electron ionization. 15, 16, 25, 36, 41, 43, 44, 92

EIC extracted ion current. 20, 21, 24, 31, 33, 34, 78, 121, 122, 124, 141

ESI electrospray ionization. 12, 14, 16

FID flame ionization detector. 11, 19, 22, 26, 34, 37, 171, 183

FT-ICR fourier transform ion cyclotron resonance. 17, 18

GMD Golm Metabolome Database. 27, 76, 106, 128

GUI graphical user interface. 131, 147

HPLC high-performance liquid chromatography. 11

IDE integrated development environment. 130, 147
IMS ion mobility spectrometry. 19
IoC inversion of control. 117

JAXB JAVA architecture for XML binding. 126

LOESS **LO**cal regr**ESS**ion. 25, 33, 36, 114, 122

MRM multiple reaction monitoring. 17
MS mass spectrometry. xiii, 2, 12, 14, 17–20, 36, 39, 41, 75, 121, 147, 171, 173
MS/MS tandem mass spectrometry. 17, 18, 20, 36, 58

NIST National Institute of Standards and Technology of the United States of America. 27, 36, 128
NMR nuclear magnetic resonance. 7, 19, 39, 148

PCA principal components analysis. 28, 76, 131, 136, 143, 183
PLS partial least squares analysis. 28
POJO plain old JAVA object. 117
PTW parametric time warping. 26
p-value probability value. 33

RCP rich client platform. 130, 131
RCS row compressed storage. 55, 113, 125
RI retention index. 76, 93, 144
RMI remote method invocation. 117, 118, 120
RPC remote procedure calling. 117
RT retention time. 33, 41, 44, 58, 67, 68

SRG seeded region growing. 122, 123, 147

TIC total ion current. 20, 21, 24, 26, 33, 34, 36, 37, 39, 40, 53, 76, 78–81, 83, 85, 86, 121–125, 134, 176, 177, 180, 181
TMS trimethylsilyl. 15, 123

TOF time-of-flight. 18, 184

URI uniform resource identifier. 112, 114

UVD UV absorbance detector. 37

XML extensible markup language. 24, 35, 36, 115, 117, 122, 126, 132, 136, 173

A

Application Examples

The following examples for GC-MS and GC×GC-MS are based on the MALTCMS framework, using the CHROMA and CHROMA4D configurations described in the previous sections. In order to run them, the latest version of MALTCMS, currently 1.3.1, needs to be downloaded and unzipped to a local folder on a computer. Additionally, MALTCMS requires a JAVA[1] runtime environment version 7 or newer to be installed. If these requirements are met, one needs to start a command prompt and change to the folder containing the unzipped MALTCMS.

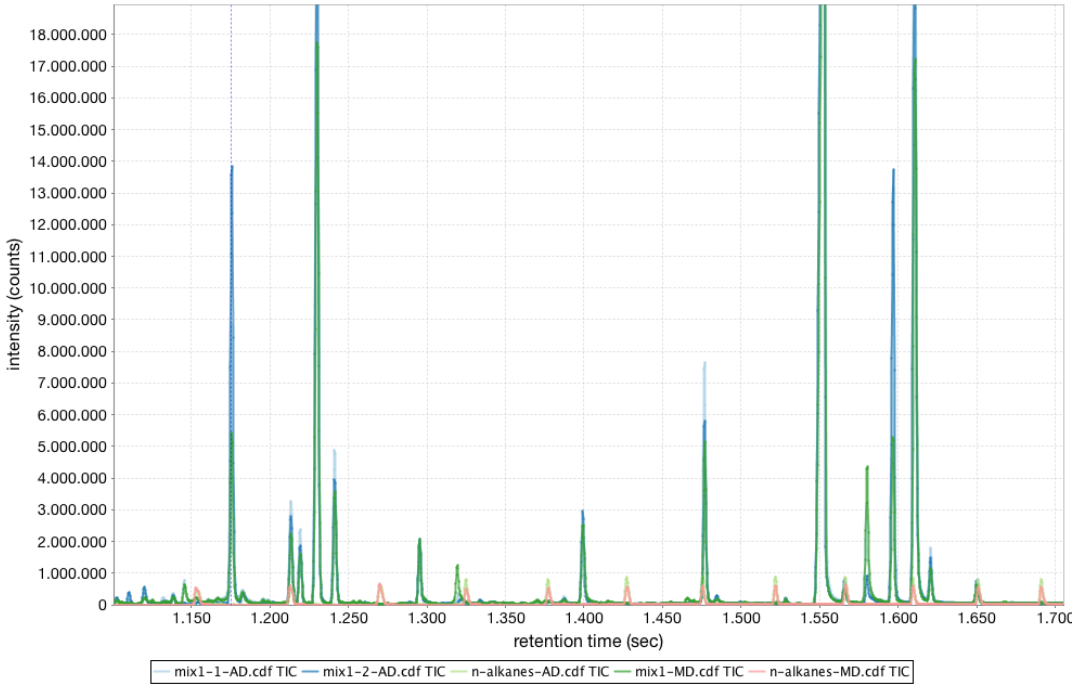
A.1. GC-MS

The experiment used to illustrate an example workflow for one-dimensional GC-MS consists of two samples of standard compounds, which contain mainly sugars, amino acids, other organic acids and nucleosides, measured after manual (MD) and after automatic derivatization (AD) with the derivatization protocol and substances given below. Group AD consists of a sample of n-alkanes standard and two replicates of mix1, namely mix1-1 and mix1-2. We will show how CHROMA can be used to find and integrate peaks, as well as compare and align the peaks between the samples, and finally how the alignment results can be used for quality control.

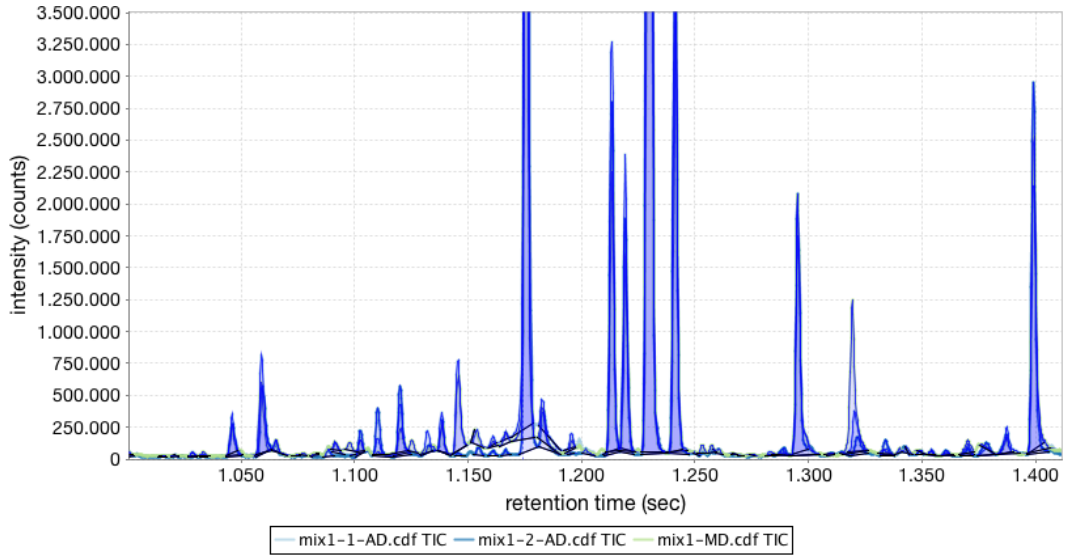
A.1.1. Sample Preparation

20 μL of each sample were incubated with 60 μL methoxylamine hydrochloride (Sigma Aldrich) in pyridine (20 mg/ml) for 90 min at 60°C before 100 μL of N-Methyl-N-(trimethylsilyl)-trifluoroacetamide (MSTFA) (Macherey & Nagel) were added for 60 min at 37°C.

1. www.java.com/



(a) Overlay of unaligned data sets, extracted from middle section within a time range of 1100 to 1700 seconds.



(b) Overlay with highlighted peak areas (without n-alkanes) after peak finding and integration. Zoomed in to provide more detail.

Figure A.1.: TIC overlay plots of the raw GC-MS data sets.

A.1.2. Acquisition and Data Processing

The samples were acquired on an Agilent GC 7890N with MSD 5975C triple axis detector (Agilent, Santa Clara CA, USA). An Agilent HP5ms column with a length of 30 m, a diameter of 0.25 mm, and a film thickness of 0.25 μm was used for the gas-chromatographic separation, followed by a deactivated restriction capillary with 50 cm length and a diameter of 0.18 mm. Per sample, 1 μL was injected onto the column in pulsed split-less mode (30 psi for 2 min). The flow rate was set to 1.5 mL/min of Helium. The linear temperature ramp started at 50 °C for 2 min until it reached its maximum of 325 °C at a rate of 10 °C/min. The raw data were exported to NetCDF format using the Agilent ChemStation software v.B.04.01 with default parameters and without additional preprocessing applied.

A sample containing n-alkanes was measured as an external standard for manual (MD) and automatic derivatization (AD) in order to be able to later determine retention indices for the other samples. The acquired data were exported to ANDI-MS (NetCDF) format before ChromA was applied. The default ChromA pipeline chroma.properties was run from the unzipped Maltcms directory with the following command (issued on a single line of input):

```
> java -Xmx1G -jar maltcms.jar -i ../data/ -o ../output/ -f *.CDF  \
 -c cfg/pipelines/chroma.mpl
```

-i points to the directory containing the input data, -o points to the directory where output should be placed, -f can be a comma separated list of filenames or, as in this case, a wildcard expression, matching all files in the input directory having a file name ending with .CDF. The final argument indicated by -c is the path to the configuration file used for definition of the pipeline and its commands. An overlay of the raw TICs of the samples is depicted in Figure A.1(a). The default ChromA pipeline configuration creates a profile matrix with nominal mass bin width. Then, the TIC peaks are located separately within each sample data file and are integrated (Figure A.1(b)). The peak apex mass spectra are then used in the next step in order to build a multiple peak alignment between all peaks of all samples by finding large cliques, or clusters of peaks exhibiting similar retention time behaviour and having highly similar mass spectra. This coarse alignment could already be used to calculate a polynomial fit, correcting retention time shift for all peaks. However, the ChromA pipeline uses the peak clusters in order to constrain a DTW alignment in the next step, which is calculated between all pairs of samples. The resulting distances are used to determine the reference sample with the lowest sum of distances to all remaining samples. Those are then aligned to the reference using the warp map obtained from the pairwise DTW calculations.

The pairwise DTW distances can easily be used for a hierarchical cluster analysis. Similar samples should be grouped into the same cluster, while dissimilar samples should be grouped into different clusters. Figure A.2 shows the results of applying a complete linkage clustering algorithm provided by the Open Source statistical

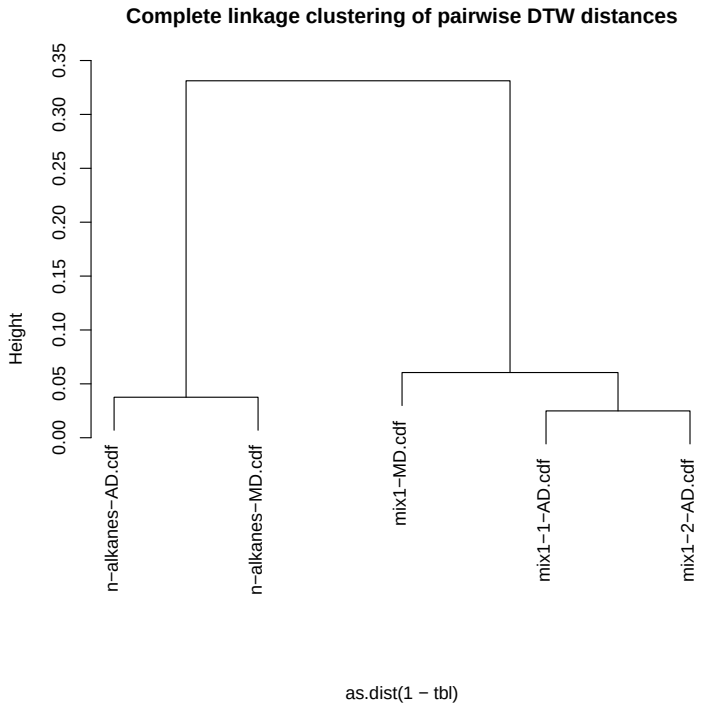


Figure A.2.: Clustering of GC-MS samples based on pairwise DTW similarities transformed to distances. The samples are clearly separated into two clusters, one containing the n-alkane standard samples, the other one containing the mix1 samples.

software R[2] to the pairwise distance matrix. It is clearly visible that the samples are grouped correctly, without incorporation of any external group assignment. Thus, this method can be used for quality control of multiple sample acquisitions, when the clustering results are compared against a pre-defined number of sample groups.

A.2. GCxGC-MS

The instructional samples presented in this section were preprocessed according to the protocol given by Doebbe et al. (2010). The description of the protocol has been adapted from that reference where necessary.

A.2.1. Sample preparation

The samples were incubated with 100 μl methoxylamine hydrochloride (Sigma Aldrich) in pyridine (20 mg/ml) for 90 min at 37°C while stirring. N- Methyl-N-

2. http://www.r-project.org

(trimethylsilyl)-trifluoroacetamide (MSTFA) (Macherey & Nagel) was then added and incubated for another 30 min at 37°C with constant stirring.

A.2.2. Acquisition and Data Processing

The sample acquisition was performed on a LECO Pegasus 4D TOF-MS (LECO, St. Joseph, MI, USA). The Pegasus 4D system was equipped with an Agilent 6890 gas chromatograph (Agilent, Santa Clara, CA, USA). The inlet temperature was set to 275°C. An Rtx-5ms (Restek, Bellefonte, PA, USA) capillary column was used with a length of 30 m, 0.25 mm diameter and 0.25 μm film thickness as the primary column. The secondary column was a BPX-50 (SGE, Ringwood, Victoria, Australia) capillary column with a length of 2 m, a diameter of 0.1 mm and 0.1 μm film thickness. The temperature program of the primary oven was set to the following conditions: 70°C for 2 min, 4°C/min to 180°C, 2°C/min to 230°C, 4°C/min to 325°C hold 3 min. This program resulted in a total runtime of about 70 min for each sample. The secondary oven was programmed with an offset of 15°C to the primary oven temperature. The thermal modulator was set 30°C relative to the primary oven and to a modulation time of 5 seconds with a hot pulse time of 0.4 seconds. The mass spectrometer ion source temperature was set to 200°C and the ionization was performed at -70eV. The detector voltage was set to 1600V and the stored mass range was 50-750 mz^{-1} with an acquisition rate of 200 spectra/second.

The raw acquired samples in LECO's proprietary ELU format were exported to NetCDF format using the LECO ChromaTOF® software v.4.22 (LECO, St. Joseph, MI, USA). Initial attempts to export the full, raw data failed with a crash beyond a NetCDF file size of 4GBytes. Thus, we resampled the data with ChromaTOF to 100 Hz (resampling factor 2) and exported with automatic signal smoothing and baseline offset correction value of 1 which resulted in file sizes around 3GBytes per sample. The samples presented in this section are named "Standard-Mix1-1" and "Standard-Mix1-2" and were measured on different days (Nov. 29th, 2008 and Dec. 12th, 2008).

The default ChromA4D pipeline for peak finding was called from within the unzipped Maltcms directory (issued on a single line of input):

```
> java -Xmx2G -jar maltcms.jar -i ../data/ -o ../output/ \
 -f *.cdf -c cfg/pipelines/chroma4D.mpl
```

The pipeline first preprocesses the data by applying a median filter followed by a top hat filter in order to remove high- and low-frequency noise contributions (Figures A.3(a) and A.3(b)). ChromA4D then uses a variant of seeded region growing in order to extend peak seeds, which are found as local maxima of the 2D-TIC. These initial seeds are then extended until the mass spectral similarity of the seed and the next evaluated candidate drops below a user-defined threshold, or until the peak area reaches its maximum, pre-defined size (Figure A.4(a)). After peak area integration, the pipeline clusters peaks between samples based on their mass spectral similarity and retention time behaviour in both dimensions to form peak cliques (not shown)

(a) 2D-TIC plot before filters were applied. Long tailing peaks are visible within the vertical dimension. Additionally, high frequency noise is present in the raw exported data, which is barely visible at this resolution.

(b) 2D-TIC plot after application of a moving median filter with window size 3 for smoothing of high-frequency noise and successive application of a top hat filter with a window size of 301 for baseline removal in order to reduce false positive peak finding results.

Figure A.3.: Visualizations of Standard-Mix1-1 before and after signal filtering with the ChromA4D processing pipeline.

(a) 2D-TIC plot of Standard-Mix1-1 after peak finding and integration with seeded region growing based on the cosine mass spectral similarity with a fusion threshold of 0.99. Peak areas were limited to contain at most 100 points.

(b) Differential plot of the two Standard-Mix1 samples after DTW alignment based on vertical TIC slices. Yellow color indicates similar amounts of total ion intensity in both samples. Green shows a surplus in Standard-Mix1-1, while red shows a surplus in Standard-Mix1-2.

Figure A.4.: Visualizations of Standard-Mix1-1 after peak finding and of Standard-Mix1-1 and Standard-Mix1-2 after alignment with DTW.

as multiple peak alignments, which are then exported into csv-format for further downstream processing. Another possible application shown in Figure A.4(b) is the visualization of pairwise GCxGC-MS alignments using DTW on the vertical 2D-TIC slices, which can be useful for qualitative comparisons.

A.3. Analytical Pyrolysis using GC-FID

MALTCMS for **A**nalytical **P**yrolysis (MALTCMS AP) is a specialized MALTCMS pipeline with a custom user interface written in Groovy. It performs TIC-based peakfinding and integration, following the method outlined in Section 5.2.3, peak area normalization to the global TIC of each chromatogram, and performs a multiple peak alignment using BIPACE RT. In contrast to the application of BIPACE RT to data acquired using a mass spectrometer as a detector, here, the detector is a pyrolysis detector, measuring a single value at each time point (see Section 2.2.1 for a short explanation). MALTCMS AP can read input data following the ANDI-CHROM conventions or from Agilent *.D* directory peak reports. It allows to execute each individual step of preprocessing, peakfinding, and peak-alignment in parallel.

MALTCMS AP is built and assembled by the *maltcms-ap-distribution* module of MALTCMS. On Unix compatible operating systems providing the Bourne-Again-Shell (Bash), it can be started by calling

```
> bin/maltcms-ap.sh
```

from the installation base directory. On Microsoft Windows systems, it can be started by invoking

```
> bin/maltcms-ap.bat
```

from the command prompt.

MALTCMS AP opens a window with a number of tabs on the left hand side of the user interface (see Figure A.5). The right hand side contains the log area used for printing the output of MALTCMS when it is running.

The import tab allows the user to select input files, either CDF files following the ANDI-CHROM convention, or CSV peak reports generated using the Agilent Mass Hunter software (Agilent, Santa Clara CA, USA), located below the chromatogram specific *.D* directories.

Depending on the choice, the user interface will selectively enable or disable certain tabs that are not applicable for the input file selection. Retention time subsetting, peakfinding and -integration are only performed if the selected files are CDF files, otherwise, only the peak-based multiple alignment using BIPACE RT is applied. The preprocessing tab allows to set a specific retention time range to be processed, or the complete chromatogram $(-\infty, \infty)$.

MALTCMS AP is a good example how MALTCMS and the components that it provides can be combined into an application that is tailored to a specific expert domain. It has been applied to compare the different amounts of lignin and carbohydrates

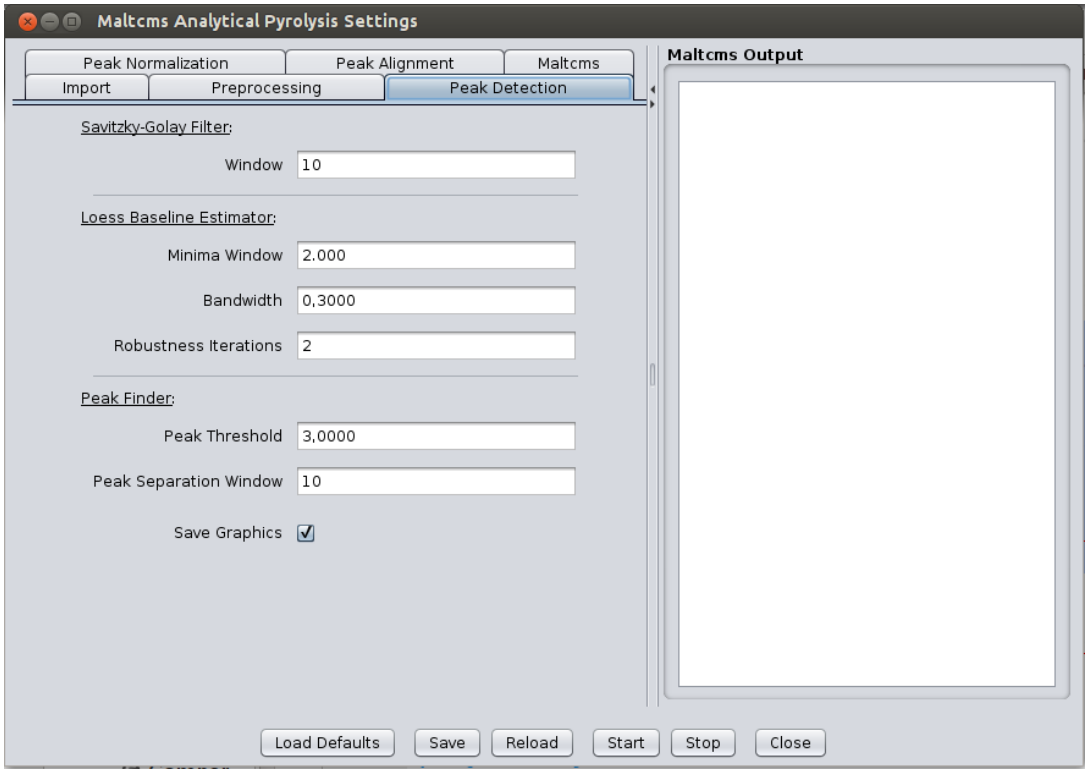


Figure A.5.: The MALTCMS AP user interface showing the tab for peak finder parameter settings.

between different transgenic variants of the genus *Populus* (poplar). The samples were acquired using a Py-2020iD micro-furnace pyrolyzer (Frontier Laboratories Ltd., Koriyama, Fukushima, Japan) mounted on an Agilent 6890 GC system. Two detectors were coupled to the GC: a flame ionization detector and an Agilent 5973 mass selective detector, using electron ionization. The experiments were conducted within the group of Dr. Dietrich Meier, Thünen Institute of Wood Chemistry, Hamburg.

The goal of this work was to find genetic traits and regulatory mechanisms that can be exploited to reduce lignin production, leading to a higher cellulose yield which is an important precursor for downstream biofuel production.

Figure A.6 shows how MALTCMS AP was used for the alignment of peaks detected using the Agilent ChemStation software and separately for the peak detection, integration, and alignment of the FID data. This preprocessing was required for the following multivariate analysis with the software UNSCRAMBLER using PCA, which showed promising results for the data-driven separation by sample origin.

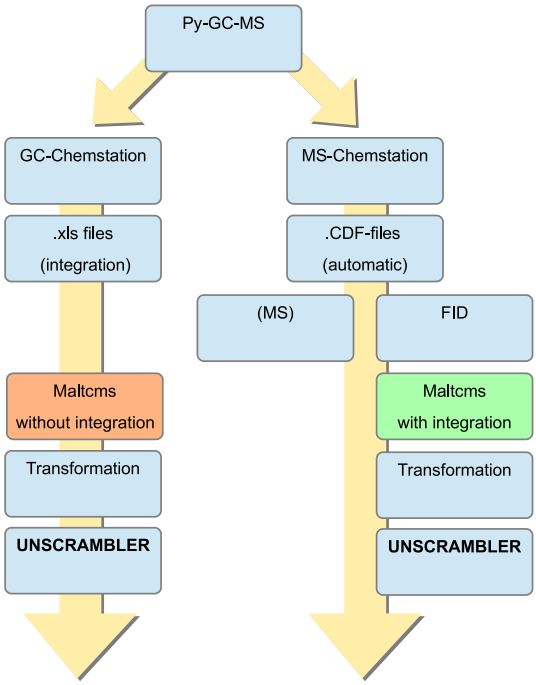


Figure A.6.: Usage workflow of MALTCMS AP as applied in poplar biofuel yield optimization. Adapted with permission from Dr. D. Meier, Hamburg.

A.4. Extension of Maui for Custom GC-MS Analysis

Based on MALTCMS and MAUI, Henning Kuich, member of the quantitative proteomics and metabolomics platform group at the Max-Delbrück-Centrum for molecular medicine in Berlin, developed customized modules and functionality to study the daily torpor in mice. Torpor is a physiological hypometabolic state that allows small animals with a high surface to volume ratio to conserve energy by lowering their core body temperature. In contrast to winter hibernation that is used also by larger mammals to conserve energy in times of fasting, the state of torpor is entered and exited much faster, allowing small animals to conserve energy throughout day and night.

Torpor is associated with fast changes in multiple vital paramaters, such as metabolic rate, heart rate, core body temperature, breathing rate, and blood pressure (Geiser 2004; Morhardt 1970; Swoap 2008).

In the torpor study the primary energy organs of the subject mice were analyzed by global metabolic profiling at six stages during the torpor process. The samples were acquired on a LECO Pegasus GC-MS TOF. Due to the number of states monitored and the variety of tissue and body liquids that were analyzed, an automated and flexible method for processing of the data was required. The required functionality was implemented based on MALTCMS and MAUI and is provided in the form of

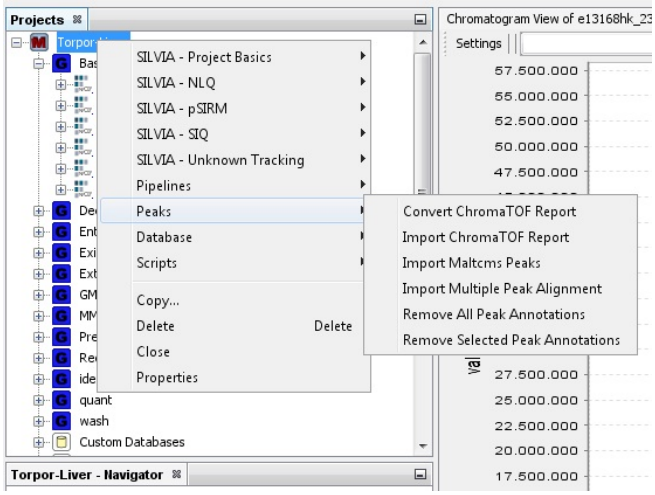


Figure A.7.: The extended MAUI user interface used during the torpor study, showing additional project level actions. Reproduced with permission from H. Kuich, Berlin.

modules that add additional actions and processing methods to the user interface (see Figure A.7).

B

Supplementary Material for BIPACE and CEMAPP-DTW

Additionally to the quantities defined in Section 3.3 (F1, Precision, Recall), we defined the following quantities to assess the coverage of a tool's reported alignment groups and associated peaks versus those contained in the reference alignment. Let $|R|$ be the number of peaks in the reference alignment and let $|T|$ be the number of peaks in a tool's reported alignment. We define

$$\text{Coverage}_R = \frac{TP + FP + FN + TN}{|R|} \tag{B.1}$$

as the fraction of peaks recovered from the alignment algorithm's reported alignment in relation to the number of peaks contained in the reference alignment. Thus, $|R| - (TP + FP + FN + TN)$ =no. of unmatched peaks in the *reference* alignment.

Equivalently, we define

$$\text{Coverage}_T = \frac{TP + FP + FN + TN}{|T|} \tag{B.2}$$

as the fraction of recovered peaks from the alignment algorithm's reported alignment in relation to the total number of peaks contained in it. Thus, $|T| - (TP + FP + FN + TN)$ =no. of unmatched peaks in the *reported* alignment. The unmatched peaks were not included in the FN values.

Ideally, Coverage_R and Coverage_T should have a value of 1, meaning that all peaks could be assigned. A low Coverage_R value may be a hint that an alignment algorithm reports too few peaks that are contained in the reference alignment, while a low Coverage_T value is a hint that an alignment algorithm is reporting many more peaks than those that are contained in the reference alignment.

Detailed tabular and graphical evaluation results for each dataset are presented in the next sections. These include figures for precision and recall values, true and false positive values (TP, FP), and true and false negative values (TN, FN). Additional figures show the algorithms' results concerning runtime and memory consumption, and the reference (Coverage_R) and tool (Coverage_T) coverage.

B.1. Result Tables

Table B.1.: Evaluation results for the *Leishmania* dataset.

Reference	Method	F1	Precision	Recall	TP	FP	TN	FN	Unm. in Ref.	Runtime (s)	Memory (MB)
Robinson	BiPACE	0.9321	0.9783	0.8901	1126	25	94	107	32	**0.24**	302.60
Robinson	BiPACE RT	0.9655	0.9933	0.9391	1188	8	111	69	8	**0.24**	**302.55**
Robinson	CeMAPP-DTW	0.9072	0.8399	0.9863	1149	219	0	**0**	16	1.30	1495.98
Robinson	CeMAPP-DTW w/ RT	0.9007	0.8289	0.9861	1134	234	0	**0**	16	1.28	1482.27
Robinson	Robinson w/ RT	**0.9976**	**0.9976**	**0.9976**	**1264**	**3**	**114**	3	**0**	—	—

Table B.2.: Evaluation results for the *Wheat* dataset. (*) MeltDB reference was generated from peaks detected by XCMS and the MeltDB profiling method.

Reference	Method	F1	Precision	Recall	TP	FP	TN	FN	Unm. in Ref.	Runtime (s)	Memory (MB)
MeltDB*	BiPACE	0.8425	0.9928	0.7317	5357	39	**0**	1924	40	3.06	**3791.16**
MeltDB*	BiPACE RT	**0.9671**	**0.9948**	0.9409	**6891**	**36**	**0**	433	**0**	**1.70**	4472.38
MeltDB*	CeMAPP-DTW	0.9344	0.9246	**0.9444**	6454	526	**0**	**380**	**0**	52.30	5233.44
MeltDB*	CeMAPP-DTW w/ RT	0.9348	0.9435	0.9263	6459	387	**0**	514	**0**	54.15	6213.91

B.2. *Leishmania* Dataset Evaluation Results

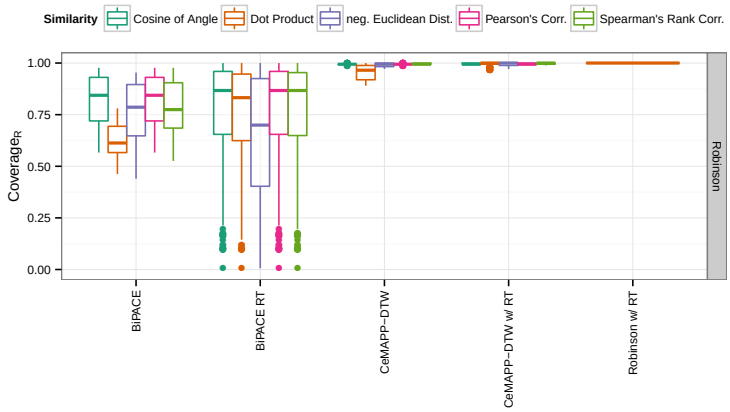


Figure B.1.: Coverage$_R$ plot for the *Leishmania* dataset. Especially the CeMAPP-DTW achieve very high coverage of the reference alignment groups, only Robinson's method performs better. BiPACE RT also has some variants that achieve coverages close to 1.

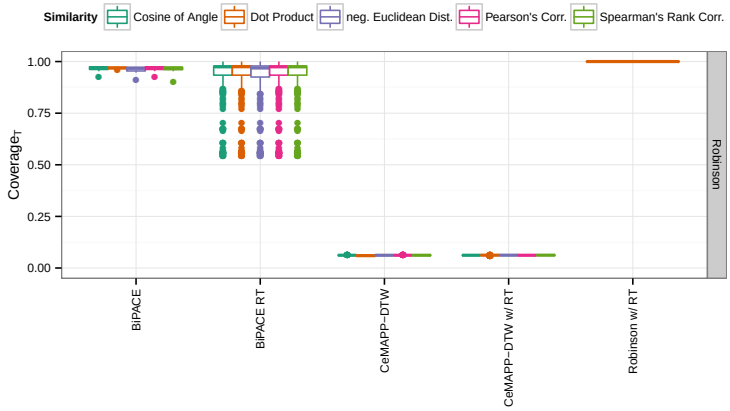


Figure B.2.: Coverage$_T$ plot for the *Leishmania* dataset. The CeMAPP-DTW variants reported many more alignment groups that were not contained in the reference alignment. Therefore, they only achieve a tool group coverage below 0.1.

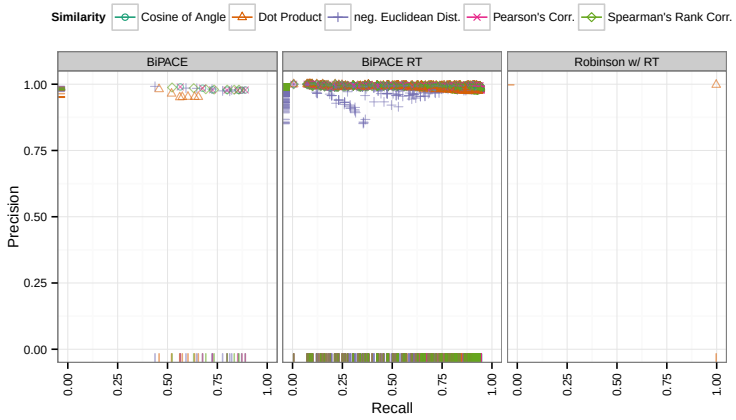


Figure B.3.: Precision and Recall plot for BiPACE for *Leishmania* dataset. Both BiPACE and BiPACE RT achieve very high precision and slightly lower recall values. However, Robinson's method is able to reach almost maximum values for either performance measure.

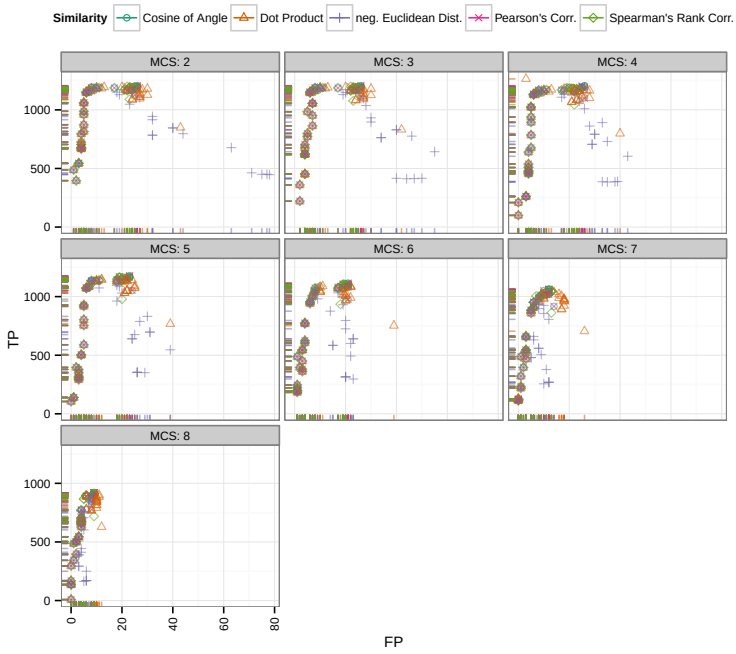


Figure B.4.: False Positives vs. True Positives for BiPACE and Robinson's method for the *Leishmania* dataset conditioned on minimum clique size (*MCS*). It is visible that the TP performance of the BiPACE variants is maximized for low *MCS* values. The minimum number of FPs is achieved for $MCS = 8$, requiring that reported cliques cover all samples.

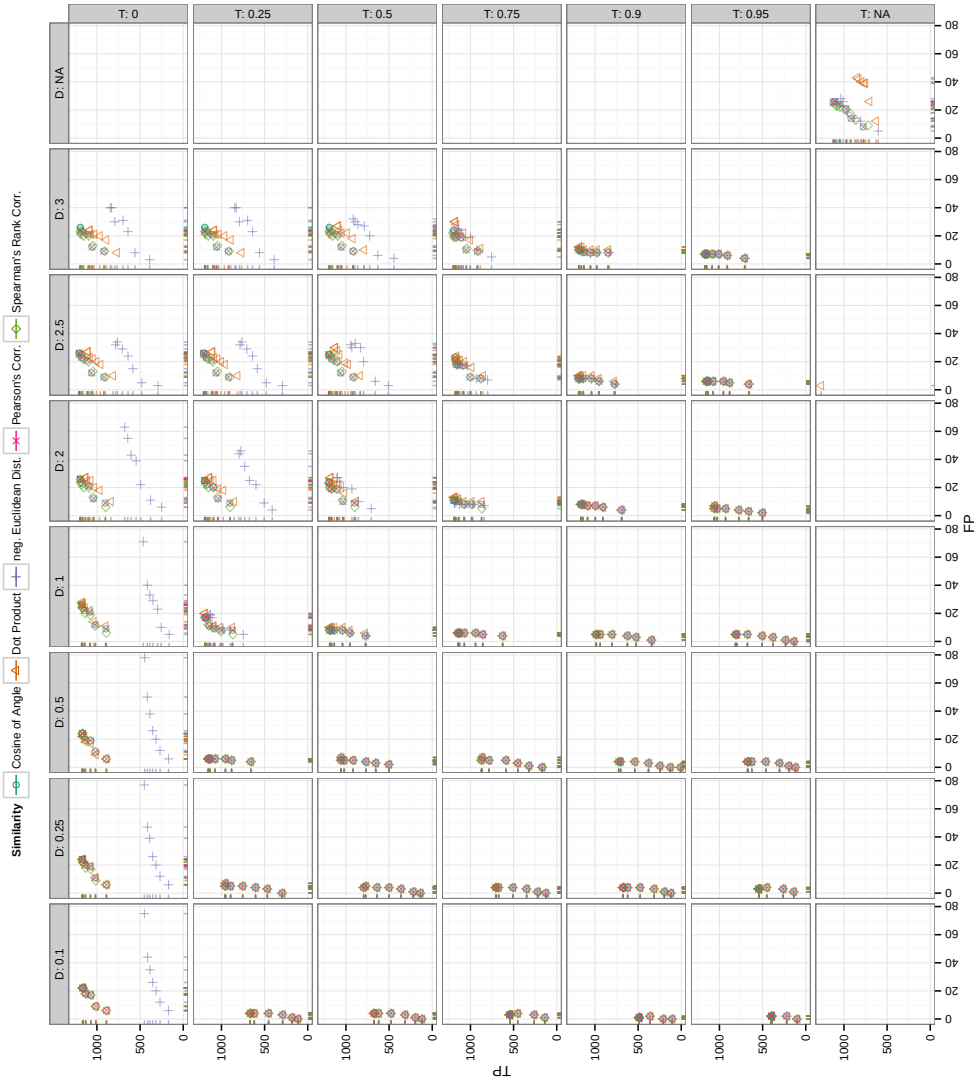


Figure B.5.: False Positives vs. True Positives for BIPACE for the *Leishmania* dataset conditioned on retention time tolerance (D) and threshold (T). BIPACE shows peaking performance close to the retention time standard deviation of the *Leishmania* dataset between $D = 2.5$ and $D = 3$ seconds. A high threshold value > 0.9 helps to drastically reduce the number of FPs.

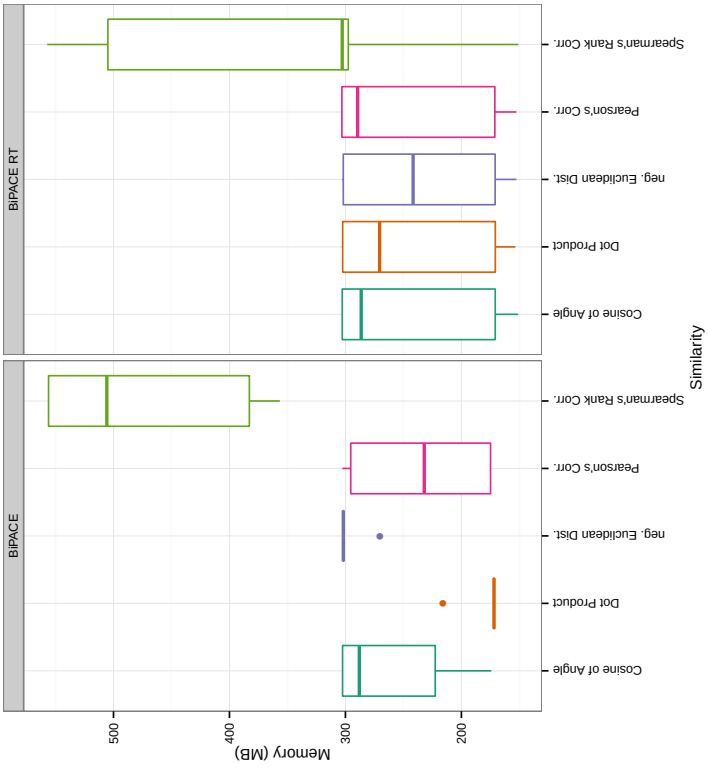


Figure B.7.: Memory plot for BIPACE for the *Leishmania* dataset. The memory requirements for Spearman's rank correlation show a much larger variation than any of the other methods, with significantly higher average memory usage.

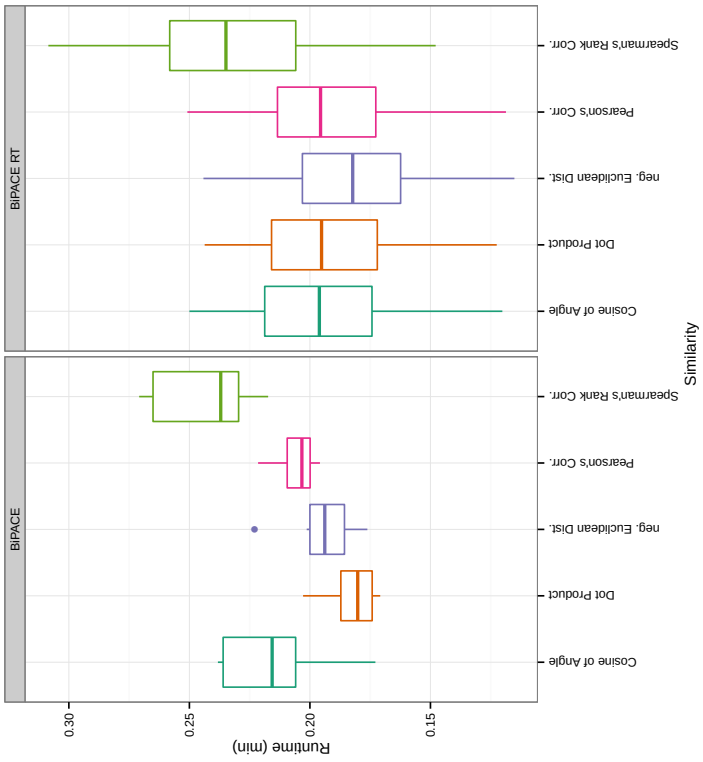


Figure B.6.: Runtime plot for BIPACE for the *Leishmania* dataset. The dot product and Euclidean distances are fastest in the computation. Longer runtimes of the BIPACE RT variants are due to the addition retention time deviation and threshold parameters.

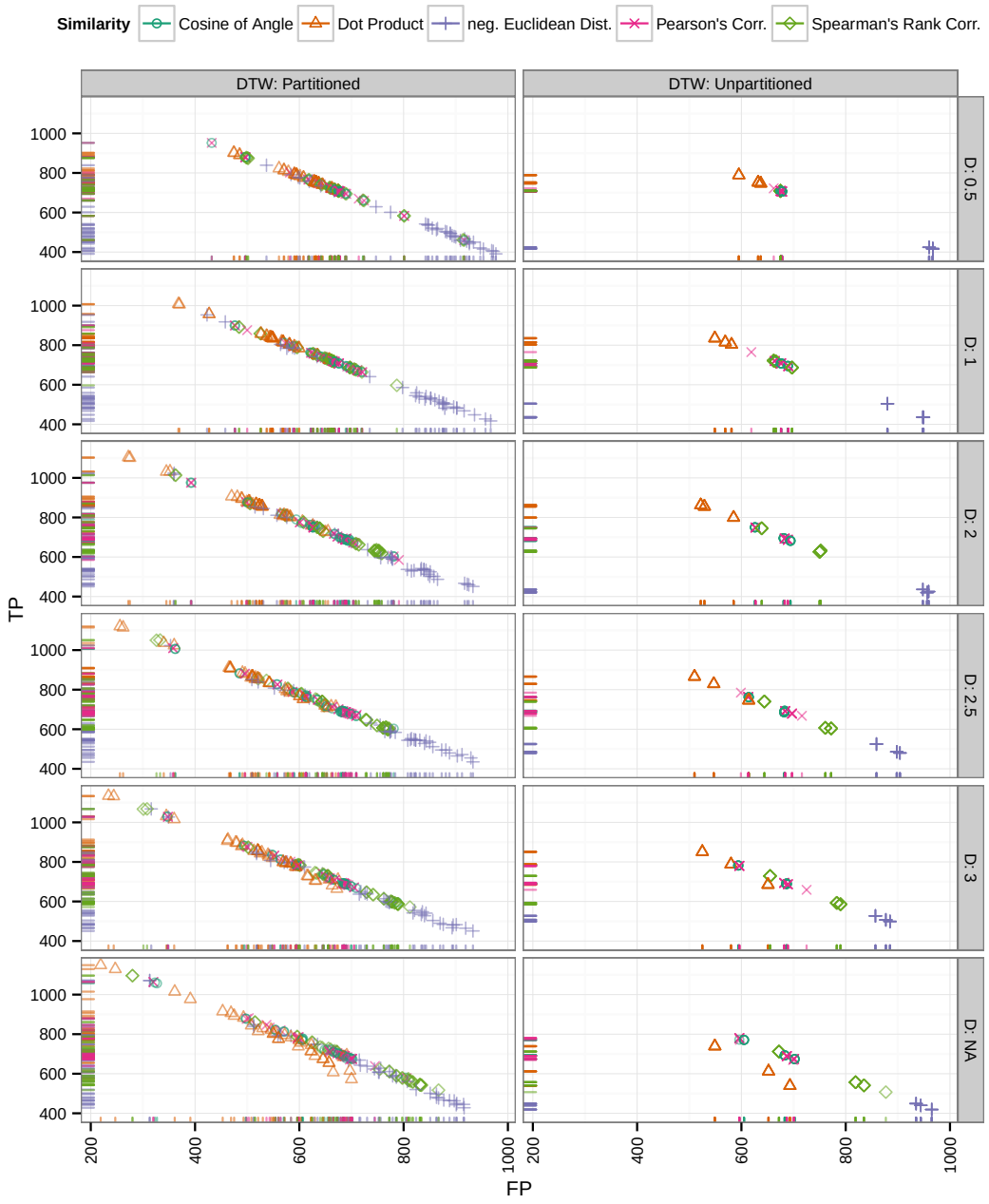


Figure B.8.: False Positives vs. True Positives for CEMAPP-DTW for the *Leishmania* dataset conditioned on partitioning and retention time tolerance (D). Partitioned instances with a large retention time deviation parameter perform consistently better. The partitioned instances that use no retention time deviation parameter perform even better, hinting at the possibility that CEMAPP-DTW does not depend on retention time information as much as BIPACE RT does.

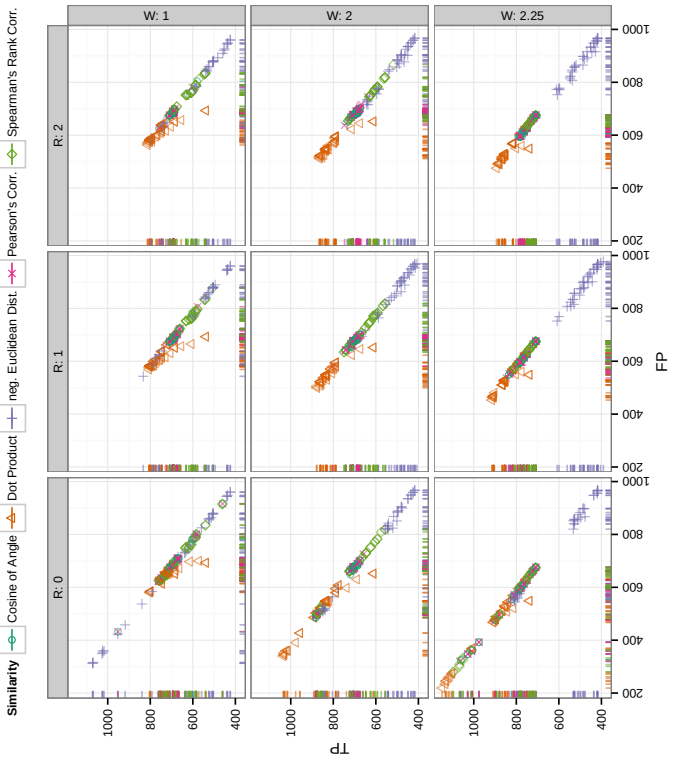


Figure B.10.: False Positives vs. True Positives for CEMAPP-DTW for the *Leishmania* dataset conditioned on anchor radius (R) and path weight (W). The combination of forcing the CEMAPP-DTW warp path through the anchors ($R = 0$) and a relaxed path weight of 2.25 achieves the best results in terms of TPs. The number of FPs is always quite high for DTW variants since they can not skip individual scans and may propose an optimal path through neighboring cells of the correct scans.

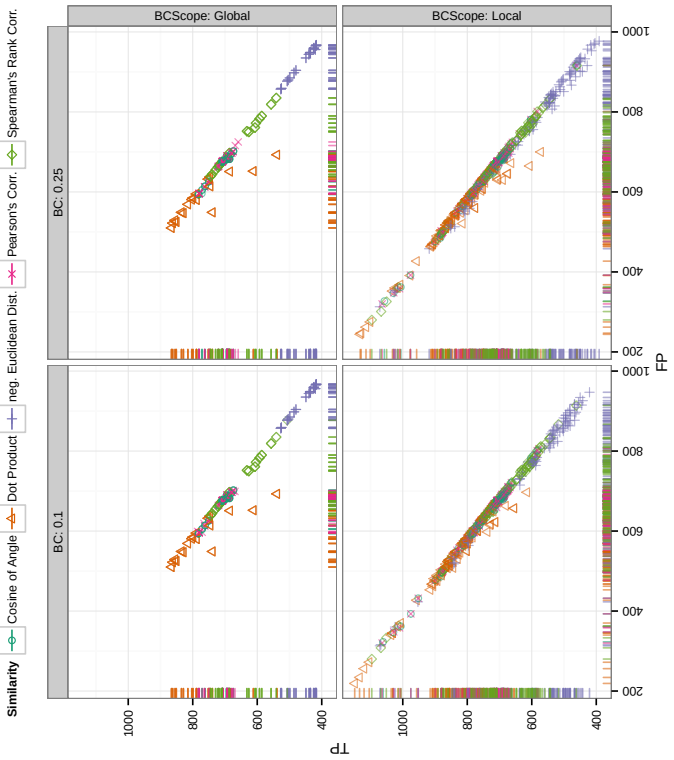


Figure B.9.: False Positives vs. True Positives for CEMAPP-DTW for the *Leishmania* dataset conditioned on relative band constraint width (BC) and scope ($BCScope$). A tight local band constraints achieves the best results (lower left panel).

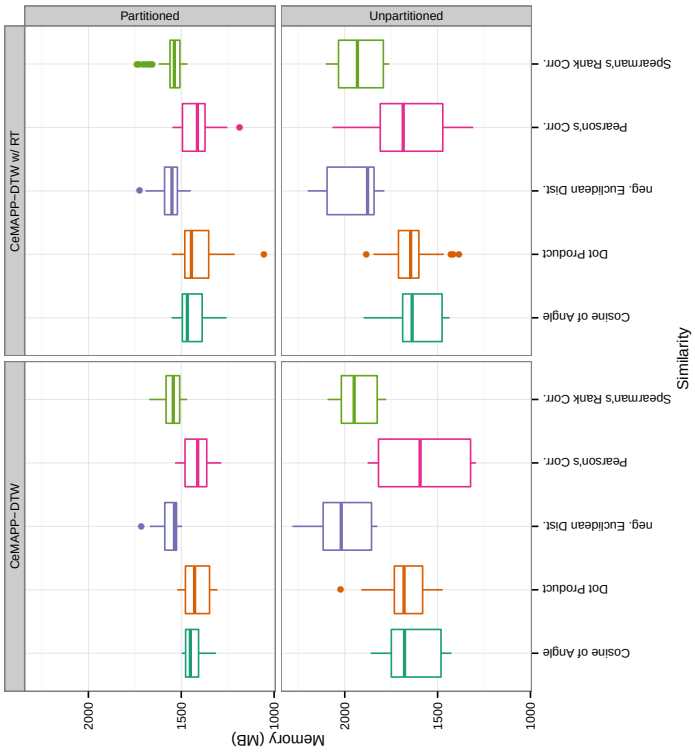


Figure B.12.: Memory plot for CeMAPP-DTW for the *Leishmania* dataset. Memory usage is reduced in partitioned versus unpartitioned DTW instances.

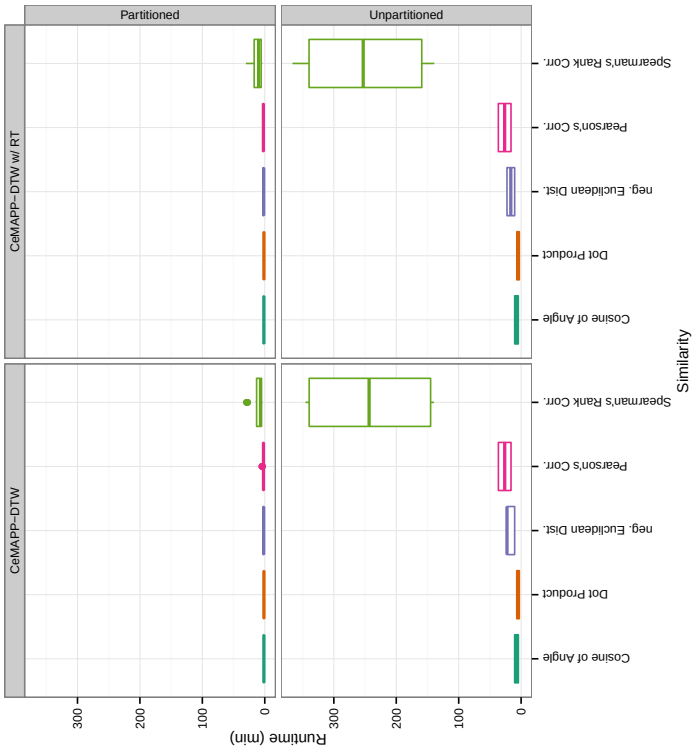


Figure B.11.: Runtime plot for CeMAPP-DTW for the *Leishmania* dataset. The partitioned variants have the lowest runtimes. Again, Spearman's rank correlation requires consistently more time to calculate than the other pairwise similarities.

B.3. *Wheat* Dataset Evaluation Results

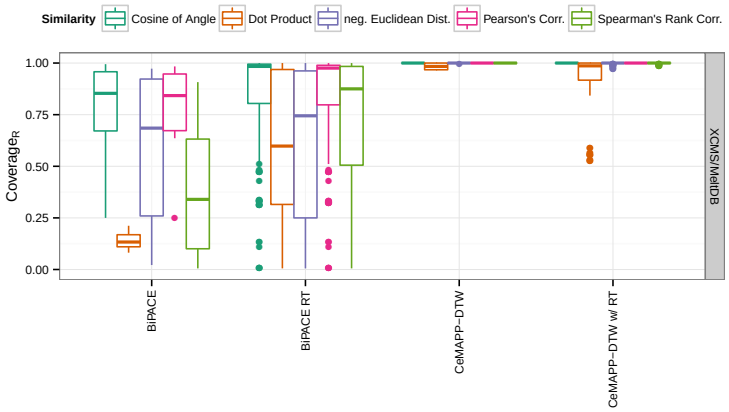


Figure B.13.: Coverage$_R$ plot for the *Wheat* dataset. The cosine and Pearson's correlation achieve very high reference alignment coverage, when BIPACE RT is used. The CEMAPP-DTW variants, as in the *Leishmania* dataset, achieve very high coverage of the reference multiple alignment.

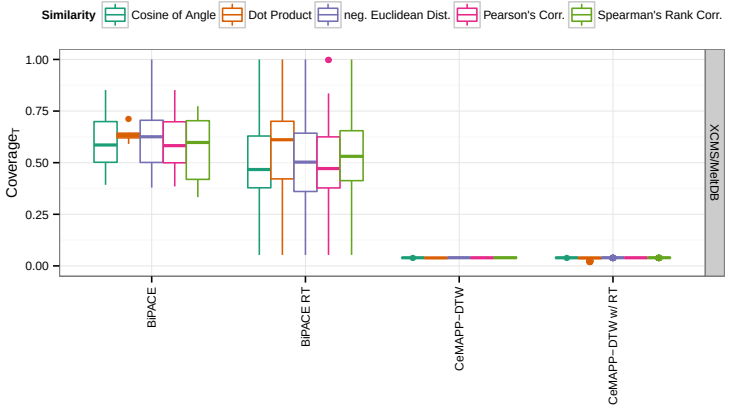


Figure B.14.: Coverage$_T$ plot for the *Wheat* dataset. The BIPACE variants can assign around 50% or more of their reported alignment groups to the reference alignment, with some instances achieving coverage values close to 100%. The CEMAPP-DTW variants can only assign well below 10% of their reported groups.

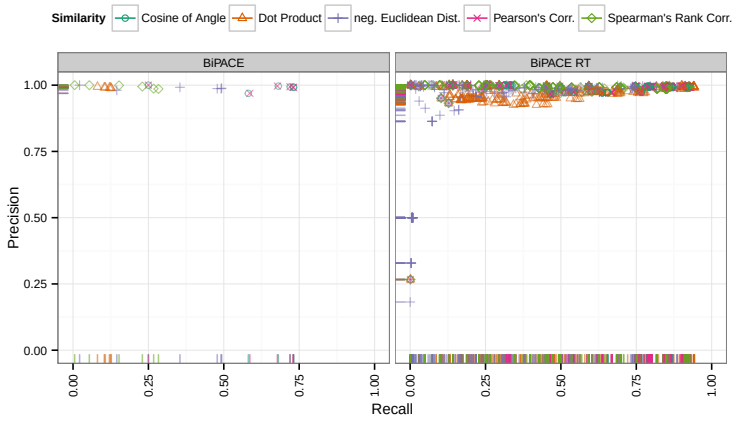


Figure B.15.: Precision and Recall plot for BIPACE for the *Wheat* dataset. BIPACE RT with the dot product as pairwise similarity clearly outperforms any other variant.

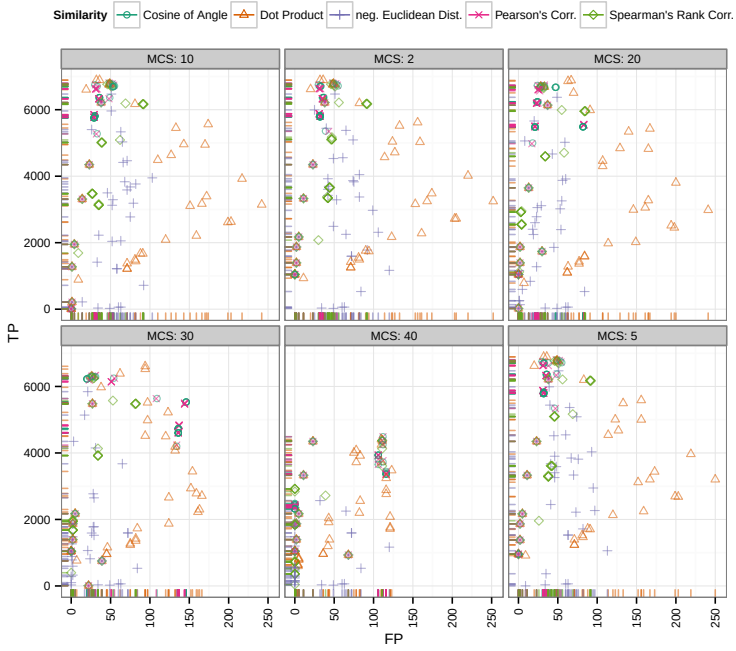


Figure B.16.: False Positives vs. True Positives for BIPACE for the *Wheat* dataset conditioned on minimum clique size (*MCS*). Here, the smallest clique size $MCS = 2$ does not differ much in the absolute number of TPs reported against the variants with $MCS = 5$ and $MCS = 10$, indicating ,that the dataset contained very similar peak groups with clear mass spectra and low retention time variance.

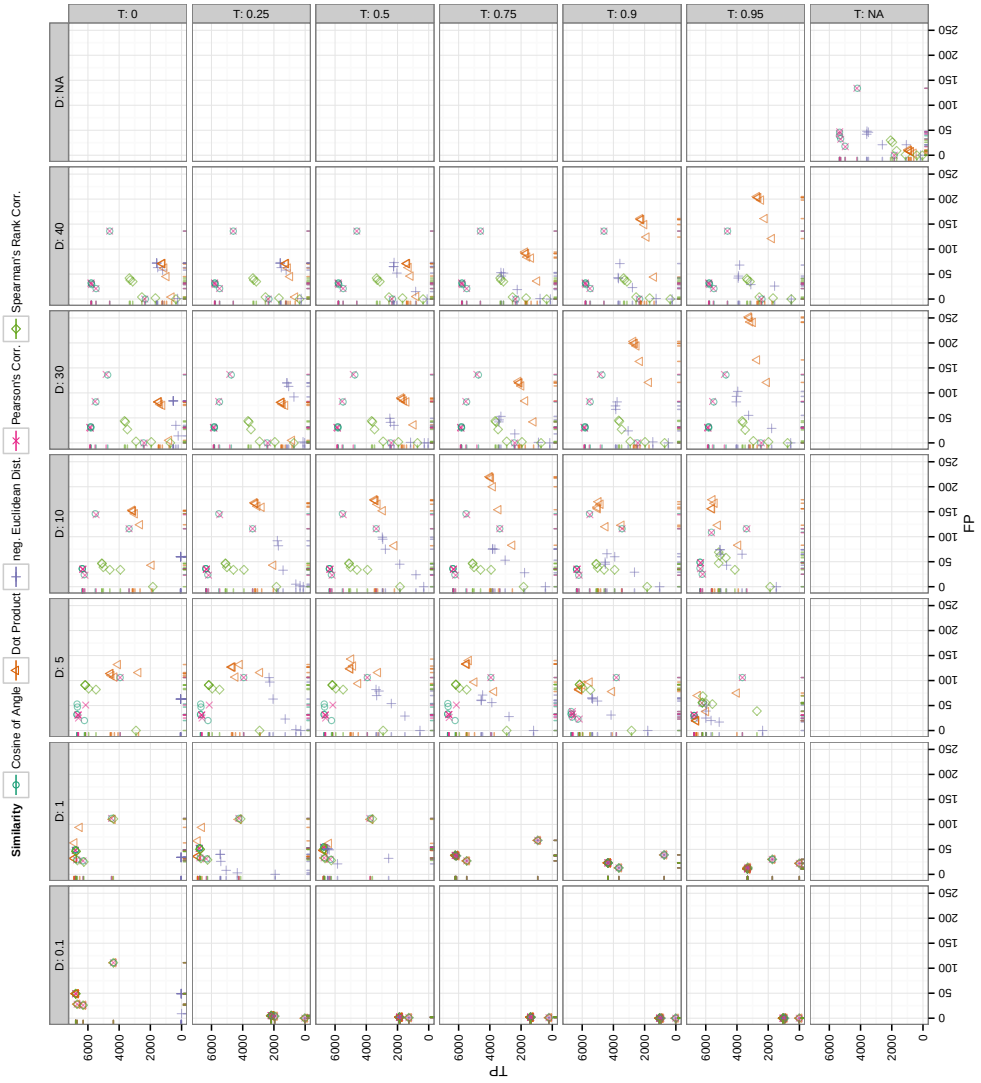


Figure B.17.: False Positives vs. True Positives for BIPACE for the *Wheat* dataset conditioned on retention time tolerance (D) and threshold (T). It is visible that the retention time tolerance parameter did not have a very large influence on the results, whereas the highest number of TPs were achieved for $T = 0$.

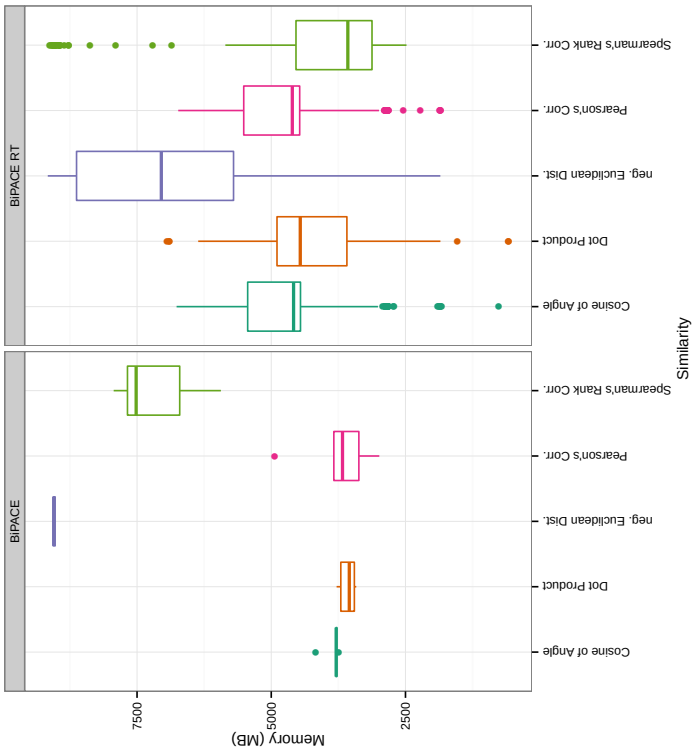


Figure B.19.: Memory plot for BIPACE for the *Wheat* dataset. Here, the negative Euclidean distance and Spearman's rank correlation require the largest amounts of memory. Spearman's rank correlation consumes less for retention time and threshold-constrained instances of BIPACE RT.

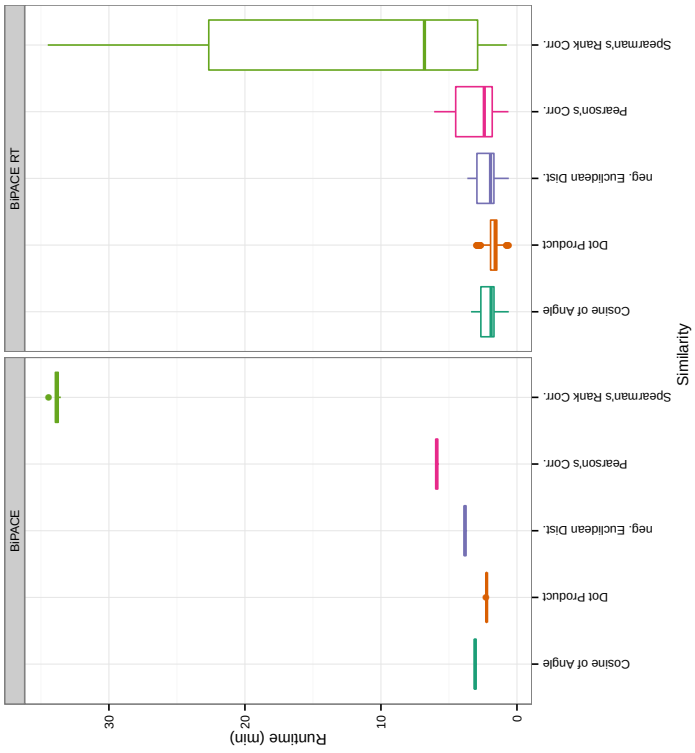


Figure B.18.: Runtime plot for BIPACE for the *Wheat* dataset. The fastest runtime was achieved for BIPACE and BIPACE RT using the dot product similarity. Spearman's rank correlation similarity was the slowest to compute.

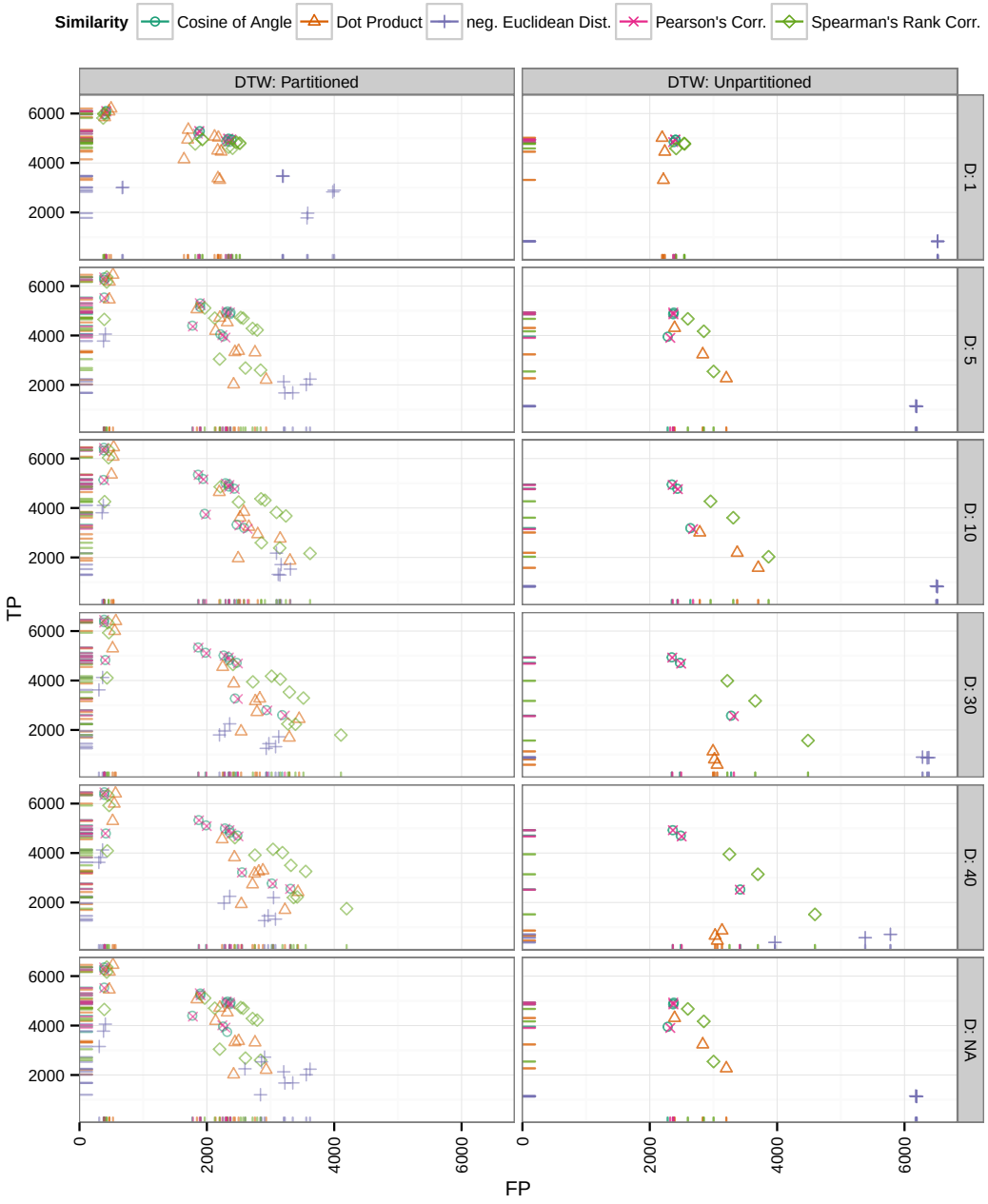


Figure B.20.: False Positives vs. True Positives of CEMAPP-DTW for the *Wheat* dataset conditioned on partitioning and retention time tolerance (D). The partitioned variants of DTW perform consistently better than the unpartitioned ones. The influence on the number of TPs and FPs decreases for increasing values of the retention time deviation parameter (D). The best results are actually obtained for the CEMAPP-DTW variants that do not use the retention time deviation and threshold parameters.

Figure B.21.: False Positives vs. True Positives of CEMAPP-DTW for the *Wheat* dataset conditioned on relative band constraint width (*BC*) and scope (*BCScope*). The locally constrained instances achieve the best results in terms of many TPs and few FPs.

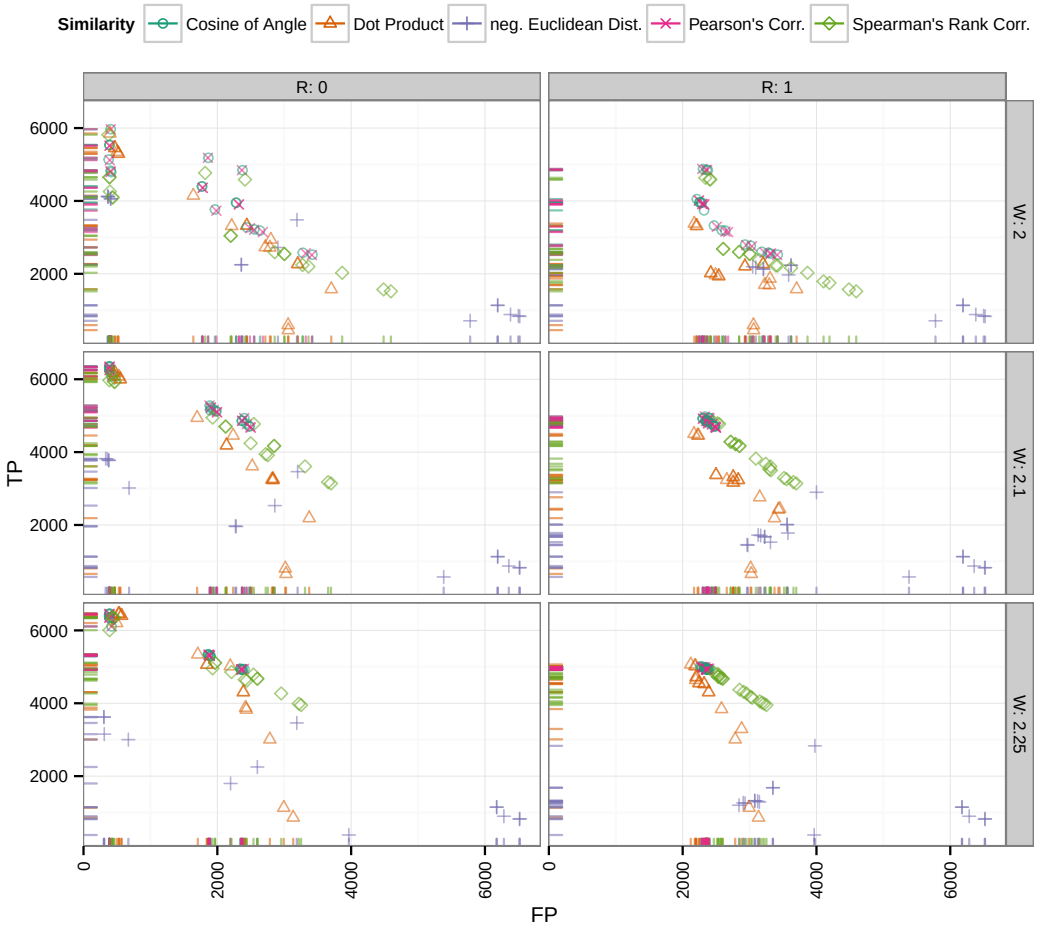
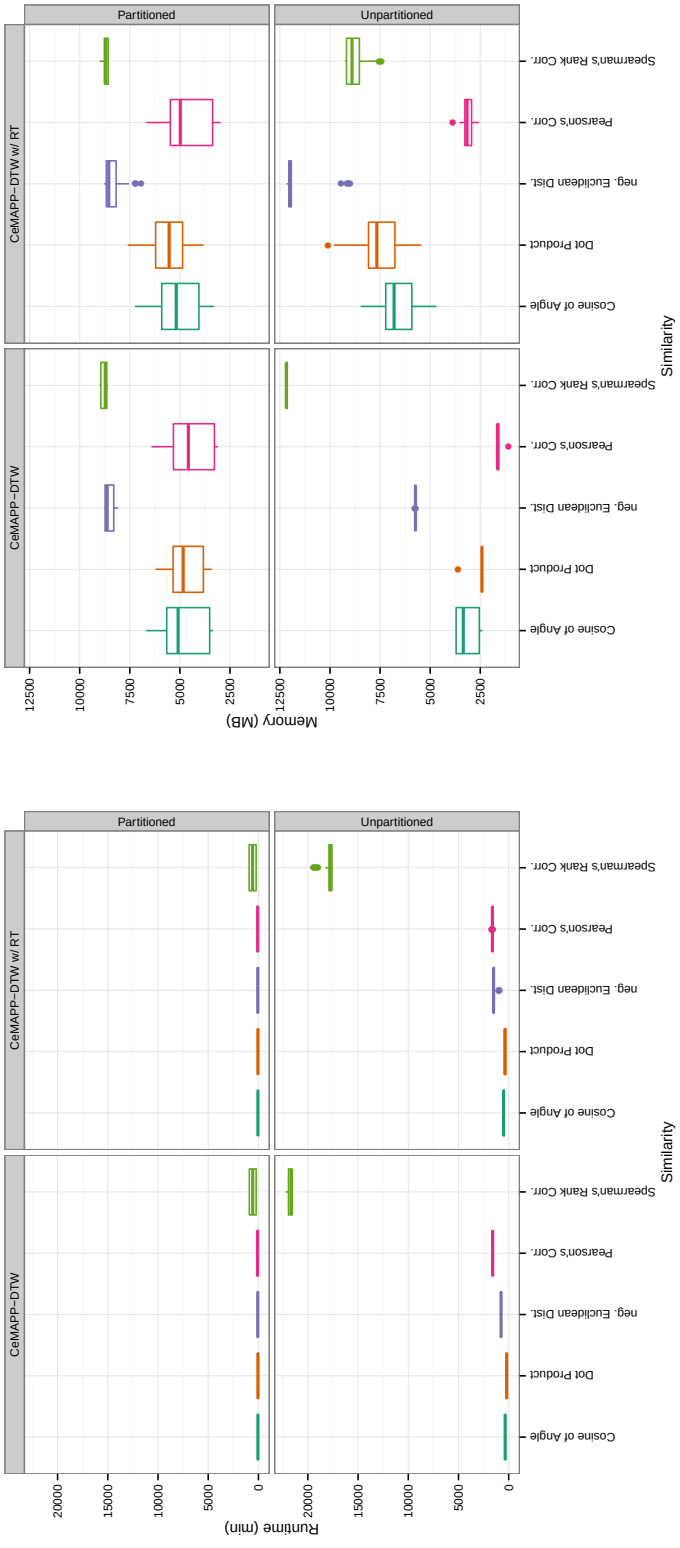


Figure B.22.: False Positives vs. True Positives of CEMAPP-DTW for the *Wheat* dataset conditioned on anchor radius (R) and path weight (W). Constraining the DTW path to the anchors ($R = 0$) in combination with a relaxed path weight ($W = 2.25$) results in the highest TP and lowest FP numbers.



(a) Runtime of CeMAPP-DTW. The partitioned variants of CeMAPP-DTW run consistently faster than the unpartitioned variants. Again, Spearman's rank correlation is the slowest pairwise similarity to compute.

(b) Memory usage of CeMAPP-DTW. The partitioned variants with retention time penalties use less memory than the unpartitioned variants, except for Pearson's correlation.

Figure B.23.: Runtime and memory plots for CeMAPP-DTW for the *Wheat* dataset.

C

Supplementary Material for BIPACE 2D

The following supplementary material compares the different reference alignment generation methods GMA and MGMA used in the main manuscript and contains further figures that show the evaluation results for all four datasets in greater detail.

We performed the following sequence of preprocessing steps to generate the automated reference multiple peak alignments:

- Elimination of duplicate peaks (same as Kim, Koo, et al. 2011; Kim, Fang, et al. 2011)
- Calculation of peak group statistics based on compound names
- Classification of peak groups using ellipses as defined by $\sigma\big(t_1(P)\big)$ and $\sigma\big(t_2(P)\big)$ parameters (see main manuscript Section 2 for details)
- Exclusion of singleton peaks and peak groups outside of ellipses
- Generation of reference multiple alignments from peak groups sorted by median retention time

We determined the parameters used for the MGMA method by examining the detailed standard deviation plots of the first $\big(\sigma(t_1)\big)$ and second column $\big(\sigma(t_2)\big)$ retention times for each peak group. We additionally considered the average, median and standard deviation of all $\sigma(t_1)$ and $\sigma(t_2)$ values. This comparison showed a minor variation of $\sigma(t_2)$ for all datasets, so that the corresponding parameter b for MGMA was generally set to 0.5. We observed a much larger variation for $\sigma(t_1)$ especially for the more complex datasets, mostly due to potentially false assignments of peak names to signals eluting far away from the majority of peaks from the corresponding peak group. Thus, we chose values for parameter a that were below the median value of $\sigma(t_1)$ measured over all peak groups.

The corresponding material for each dataset is available in Supplementary File 2 of Hoffmann et al. (2014). This includes details on every peak group for each dataset including boxplots of the individual variation in first and second dimension retention times, grouped by peak name and chromatogram file. Additionally, the same file

contains plots of the standard deviations of every peak group for the different datasets along with the decision boundary and a graphical indication, whether a group is considered an outlier group, or not, for easier comprehension of the tabular data that is also supplied within the same file.

We then compared and visualized the peak sets and their assigned names (encoded as plot symbol and color) for each of the four references (Figures C.1–C.4).

Additionally to the quantities defined in the main manuscript (F1, Precision, Recall), we used the Coverage_R and Coverage_T quantities as defined in Appendix B to assess the coverage of a tool's reported alignment groups and associated peaks versus those contained in the reference alignment.

A discussion comparing the advantages and disadvantages of pairwise alignment performance evaluation, as used by Kim *et al.* (Kim, Koo, et al. 2011; Kim, Fang, et al. 2011), and our row-wise multiple alignment evaluation is given in Section C.7. Results of the pairwise evaluation are given in the respective sections for each dataset.

C.0.1. Structure

The remainder of the supplementary material is structured as follows: In Section C.1, we show figures comparing the peak sets of the GMA and MGMA reference alignments, illustrating the need for improved filtering of peak groups, as performed by MGMA, due to mis-assignments when relying solely on the maximum peak area as a peak group assignment criterion. Additionally, these figures show the differences between the reference multiple alignments side by side for easier visual perception than the tabular data provided in Supplementary File 2 of Hoffmann et al. (2014).

Detailed evaluation results for each dataset are presented in Sections C.2–C.5. These include summary tables of the best parametrizations, categorized by algorithm and reference alignment, as well as figures for the average pairwise F1 score F1_p, as introduced by Kim *et al.*, Precision and Recall values, true and false positive values (TP, FP), and true and false negative values (TN, FN). Additional figures show the algorithms' results concerning runtime and memory consumption, and the reference (Coverage_R) and tool (Coverage_T) coverage. All original tables and figures are contained in Supplementary File 2 of Hoffmann et al. (2014).

In Section C.6, we give an overview of the parameters used in BiPACE 2D and how to set up and configure the version of BiPACE 2D supplied with the framework MaltCMS for GC×GC-MS data.

Section C.7 contains a short discussion about the advantages and disadvantages of the pairwise evaluation in contrast to our evaluation method.

C.1. Comparison of GMA and MGMA Reference Alignments

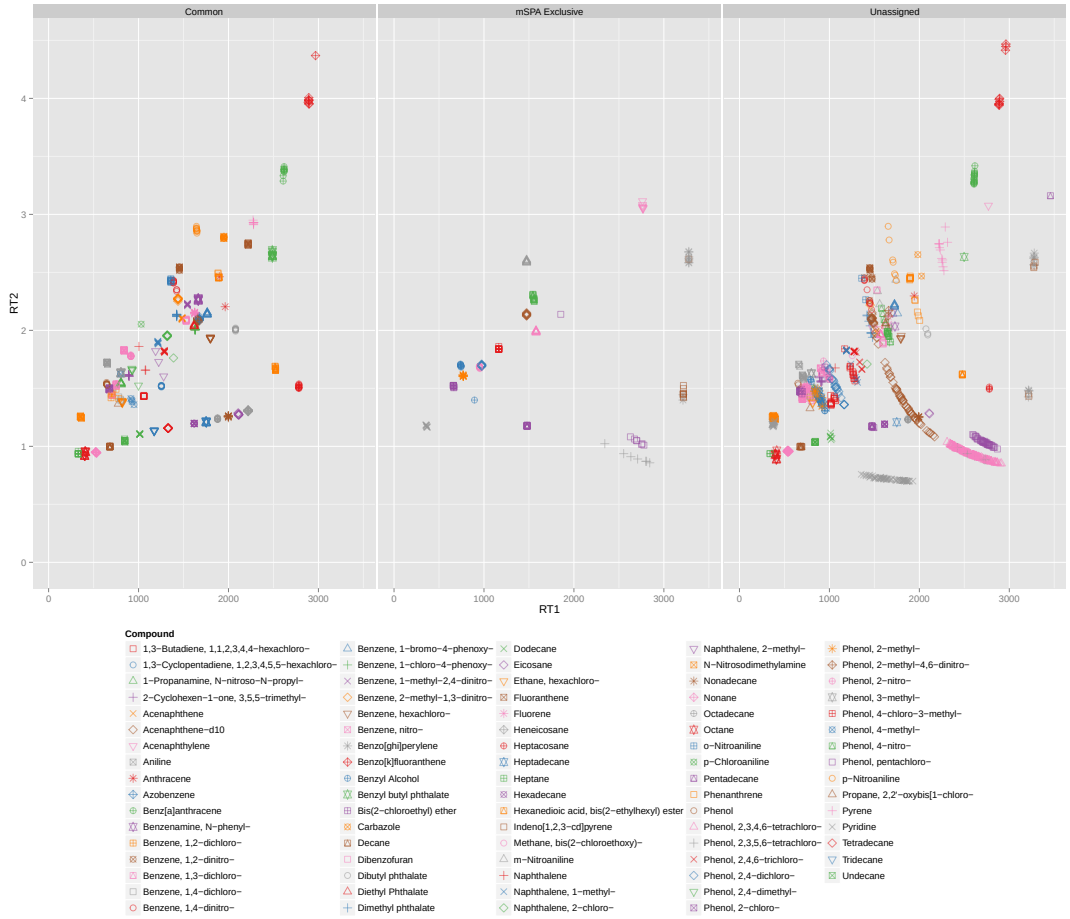


Figure C.1.: Depiction of the peak sets of mSPA dataset I used for automatic multiple alignment reference generation. Peaks with the assumed same identity across peak reports share the same color and shape. The individual facets show, from left to right, the common peaks (equivalent to the MGMA reference), peaks unique to the original approach used in the mSPA publication (termed GMA in the main manuscript), and the peaks that were not selected as references (unassigned). A total of 17 out of 83 peak groups were removed from further consideration if they had standard deviations of $\sigma(t_1) > 50$ and $\sigma(t_2) > 0.5$ for the first and second column retention times. Details are available in Supplementary File 2 below mSPA dataset I within the tabular file *compoundGroupStatsAll.txt*.

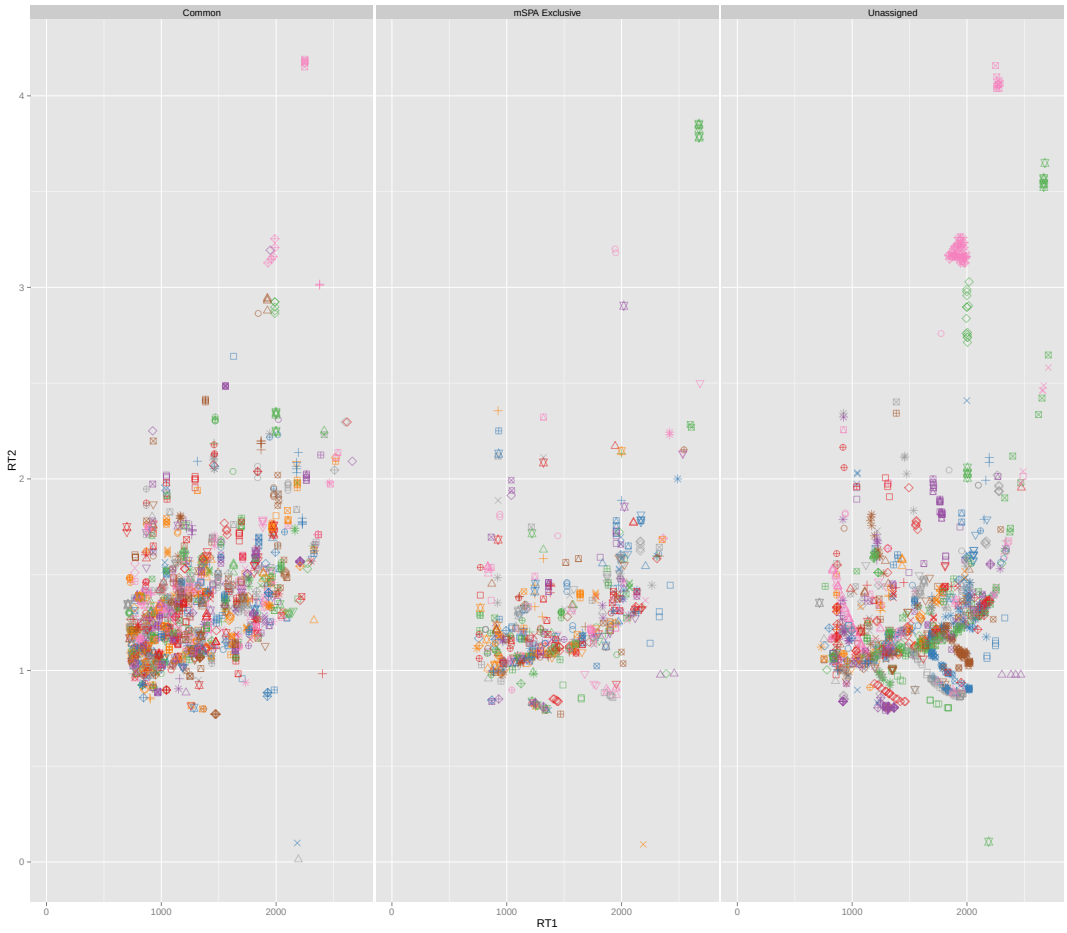


Figure C.2.: Depiction of the peak sets of mSPA dataset II used for automatic multiple alignment reference generation. Peaks with the assumed same identity across peak reports share the same color and shape. The individual facets show, from left to right, the common peaks (equivalent to the MGMA reference), peaks unique to the original approach used in the mSPA publication (termed GMA in the main manuscript), and the peaks that were not selected as references (unassigned). Peak names had to be omitted for this figure's legend, but are available in Supplementary File 2 below mSPA dataset II within the tabular file *compoundGroupStatsAll.txt*. A total of 182 out of 1039 peak groups were removed from further consideration if they had standard deviations of $\sigma(t_1) > 55$ and $\sigma(t_2) > 0.5$ for the first and second column retention times.

Figure C.3.: Depiction of the peak sets of SWPA dataset I used for automatic multiple alignment reference generation. Peaks with the assumed same identity across peak reports share the same color and shape. The individual facets show, from left to right, the common peaks (equivalent to the MGMA reference), peaks unique to the original approach used in the MSPA publication (termed GMA in the main manuscript), and the peaks that were not selected as references (unassigned). Peak names had to be omitted for this figure's legend, but are available in Supplementary File 2 below SWPA dataset I within the tabular file *compoundGroupStatsAll.txt*. A total of 8 out of 84 peak groups were removed from further consideration if they had standard deviations of $\sigma(t_1) > 800$ and $\sigma(t_2) > 0.5$ for the first and second column retention times.

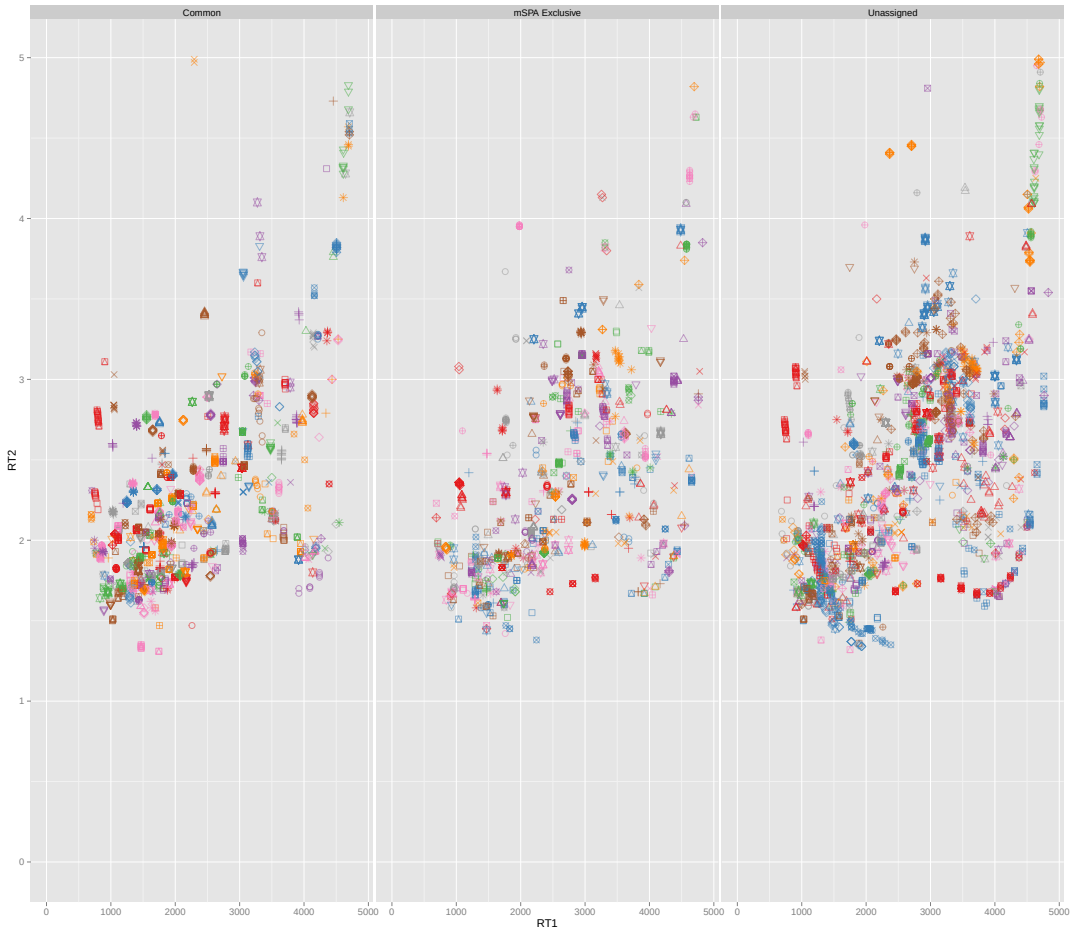


Figure C.4.: Depiction of the peak sets of CHLAMY dataset I used for automatic multiple alignment reference generation. Peaks with the assumed same identity across peak reports share the same color and shape. The individual facets show, from left to right, the common peaks (equivalent to the MGMA reference), peaks unique to the original approach used in the MSPA publication (termed GMA in the main manuscript), and the peaks that were not selected as references (unassigned). A total of 146 out of 449 peak groups were removed from further consideration if they had standard deviations of $\sigma(t_1) > 250$ and $\sigma(t_2) > 0.5$ for the first and second column retention times. Further details are available in Supplementary File 2 below CHLAMY dataset I within the tabular file *compoundGroupStatsAll.txt*

C.2. MSPA Dataset I Evaluation Results

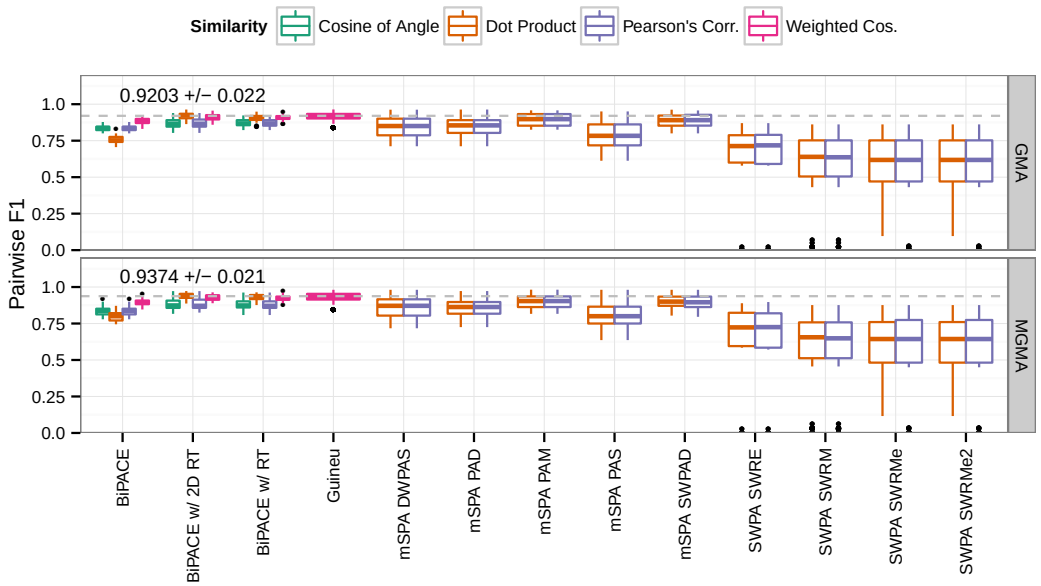


Figure C.5.: Best pairwise average F1 instances for MSPA dataset I. BIPACE 2D with the cosine score achieves the highest average pairwise values and has the lowest standard deviation.

Table C.1.: Best evaluation results for MSPA Dataset I for each algorithm variant concerning achieved F1 score on the GMA and MGMA references. The best value within each column is highlighted in bold face. BIPACE 2D achieves the highest F1 and Precision scores for either reference and also has the lowest memory usage. GUINEU achieves the highest Recall value and lowest runtime.

Reference	Method	F1	Precision	Recall	TP	FP	TN	FN	Unm. in Ref.	Runtime (s)	Memory (MB)
GMA	BiPACE	0.9018	0.9477	0.8602	634	35	38	73	30	2.98	76.92
GMA	BiPACE w/ 2D RT	**0.9296**	**0.9551**	0.9053	**660**	**31**	**50**	49	20	3.19	**60.66**
GMA	BiPACE w/ RT	0.9191	0.9519	0.8884	653	33	42	62	20	3.41	74.02
GMA	Guineu	0.9082	0.8701	**0.9498**	643	96	37	**24**	10	**2.74**	103.72
GMA	mSPA DWPAS	0.8858	0.9199	0.8541	609	53	44	104	**0**	26.30	86.30
GMA	mSPA PAD	0.8824	0.9075	0.8588	608	62	40	90	10	24.54	86.30
GMA	mSPA PAM	0.9049	0.9378	0.8743	633	42	44	81	10	27.14	86.30
GMA	mSPA PAS	0.8488	0.9000	0.8031	567	63	41	129	10	21.11	86.30
GMA	mSPA SWPAD	0.9004	0.9429	0.8615	628	38	43	91	10	42.25	86.30
GMA	SWPA SWRE	0.7248	0.8865	0.6130	453	58	13	176	110	11.48	86.40
GMA	SWPA SWRM	0.6972	0.8436	0.5941	426	79	14	171	120	12.43	86.40
GMA	SWPA SWRMe	0.6957	0.8360	0.5958	423	83	17	187	100	12.21	86.40
GMA	SWPA SWRMe2	0.6957	0.8360	0.5958	423	83	17	187	100	12.50	86.40
MGMA	BiPACE	0.9117	0.9481	0.8780	511	28	30	51	20	2.98	76.92
MGMA	BiPACE w/ 2D RT	**0.9472**	**0.9607**	0.9340	**538**	**22**	**42**	28	10	3.19	**60.66**
MGMA	BiPACE w/ RT	0.9322	0.9549	0.9105	529	25	34	42	10	3.41	74.02
MGMA	Guineu	0.9165	0.8731	**0.9645**	516	75	30	**19**	0	**2.74**	103.72
MGMA	mSPA DWPAS	0.8952	0.9212	0.8706	491	42	34	73	**0**	26.30	86.30
MGMA	mSPA PAD	0.8925	0.9193	0.8673	490	43	32	65	10	24.54	86.30
MGMA	mSPA PAM	0.9169	0.9361	0.8984	513	35	34	48	10	24.49	86.30
MGMA	mSPA PAS	0.8606	0.9182	0.8099	460	41	31	98	10	21.11	86.30
MGMA	mSPA SWPAD	0.9102	0.9441	0.8787	507	30	33	60	10	42.25	86.30
MGMA	SWPA SWRE	0.7405	0.8771	0.6408	371	52	9	128	80	11.48	86.40
MGMA	SWPA SWRM	0.7043	0.9270	0.5679	343	27	9	191	70	11.60	86.40
MGMA	SWPA SWRMe	0.7065	0.8448	0.6071	343	63	12	142	80	12.21	86.40
MGMA	SWPA SWRMe2	0.7065	0.8448	0.6071	343	63	12	142	80	12.50	86.40

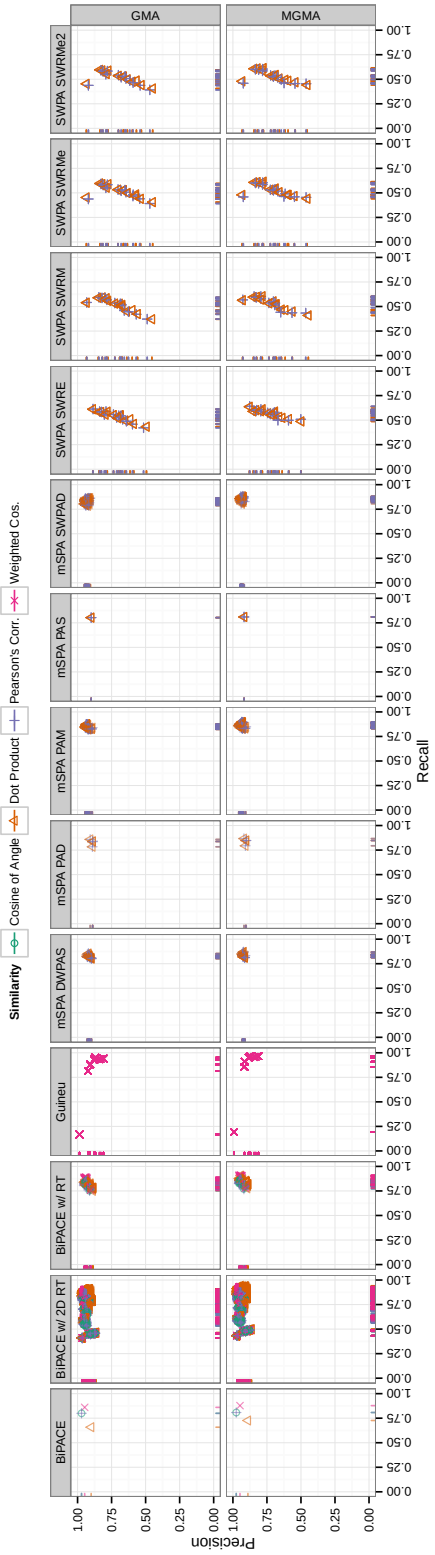


Figure C.6.: Precision and Recall plot for MSPA dataset I. BIPACE 2D achieves the highest Precision and Recall values on both references, using the dot product as mass spectral similarity. MSPA-DWPAS has the highest Precision values among the other evaluated methods, also for both references, while MSPA-PAM has the highest Recall values. The choice of dot product or Pearson's linear correlation for the MSPA and SWPA variants seems to have a rather small influence.

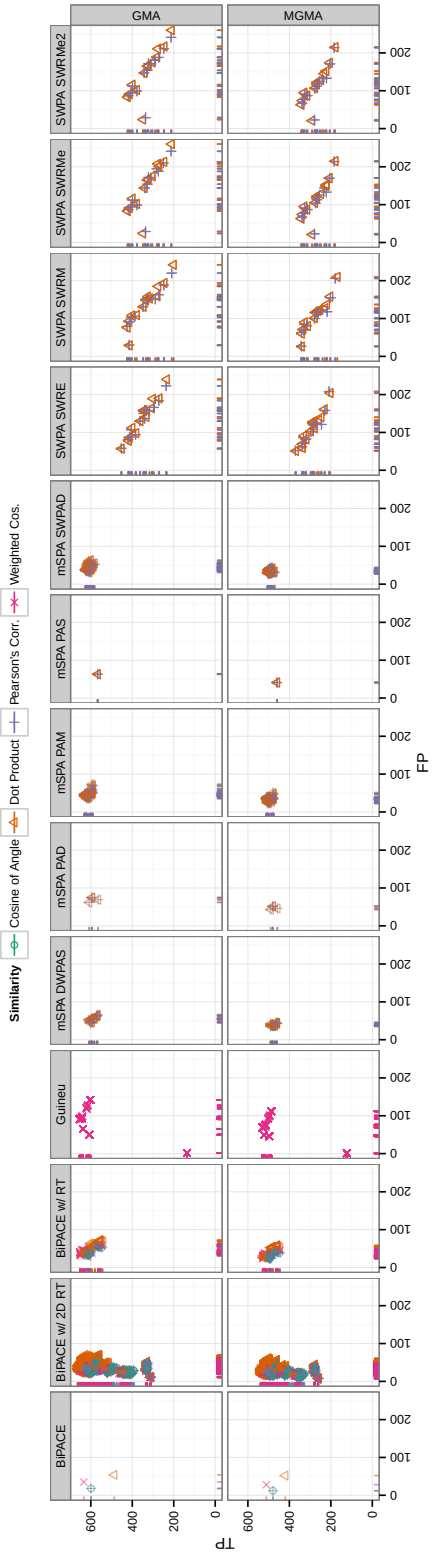


Figure C.7.: False Positives vs. True Positives for MSPA dataset I. BIPACE 2D performs best, with the lowest number of false positives, while achieving the highest number of true positives on either reference. BIPACE RT also performs comparably well, with fewer true positives. Of the other methods, MSPA-PAM performs best, using the dot product as pairwise mass spectral score, followed closely by MSPA-SW-PAD. GUINEU achieves very high TP values, but at the cost of a high number of FP values.

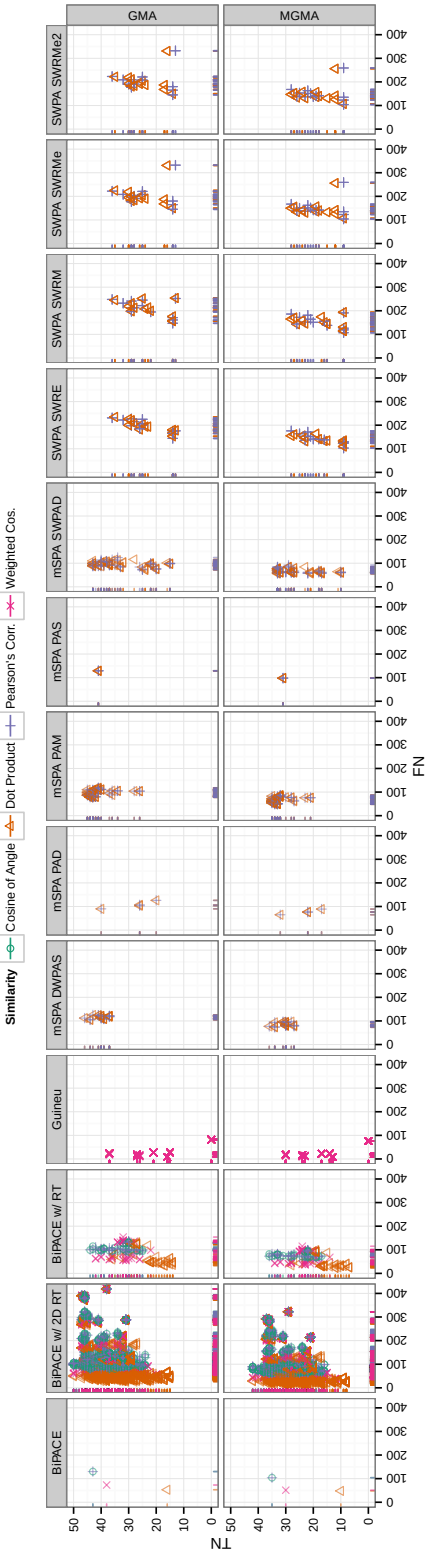


Figure C.8.: False Negatives vs. True Negatives for MSPA dataset I. All three BIPACE variants achieve higher true negatives than any of the MSPA or SWPA variants for many instances and comparatively low false negative number for some instances. GUINEU achieves consistently low FN values.

Figure C.9.: Runtime plot for mSPA dataset I. The BiPACE variants all vary between 3 and 7 seconds in runtime, while the mSPA variants vary between 20 and over 30 seconds. The SWPA variants have competetive runtimes around 10 seconds. Guineu has the fastest runtimes below 3 seconds.

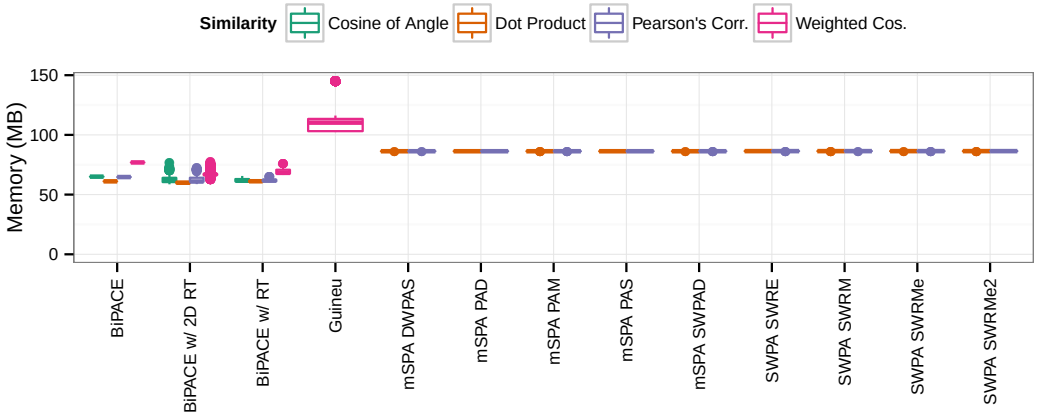


Figure C.10.: Memory plot for mSPA dataset I. The BiPACE variants consume between 60-85 MBytes of memory. The mSPA and SWPA variants consume around 80 MBytes of memory. Guineu consumes most at around 100-110 MBytes.

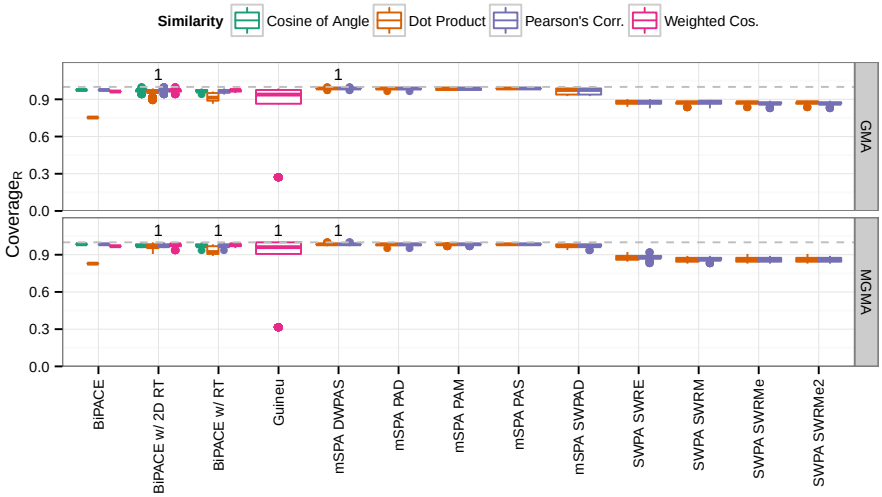


Figure C.11.: Coverage$_R$ plot for mSPA dataset I. The majority of variants achieve coverage values of $> 90\%$ of the reference peaks, meaning that they reported only a small fraction of peaks that were not assignable to the reference alignment. The SWPA variants fall behind with $> 85\%$ coverage.

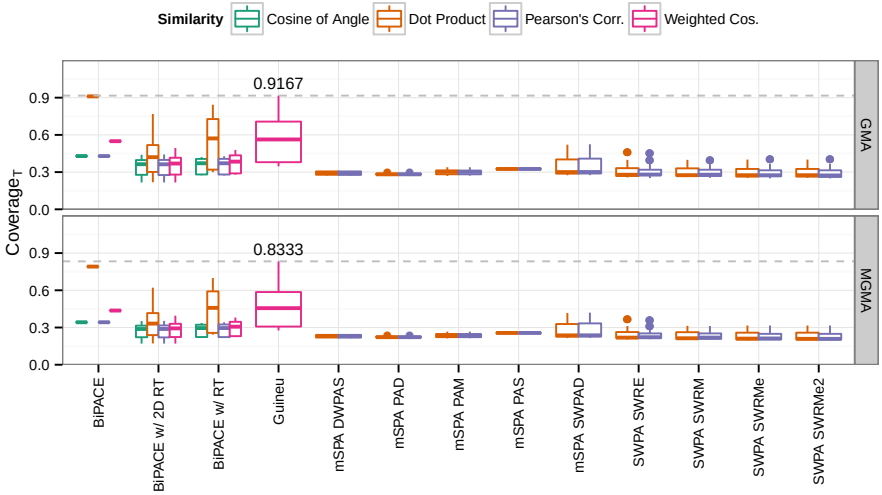


Figure C.12.: Coverage$_T$ plot for mSPA dataset I. Most instances can assign between 25 and 30% of their own reported peak groups to the reference alignment, except for Guineu, which has a much higher median value and assigns over 80%.

C.3. mSPA Dataset II Evaluation Results

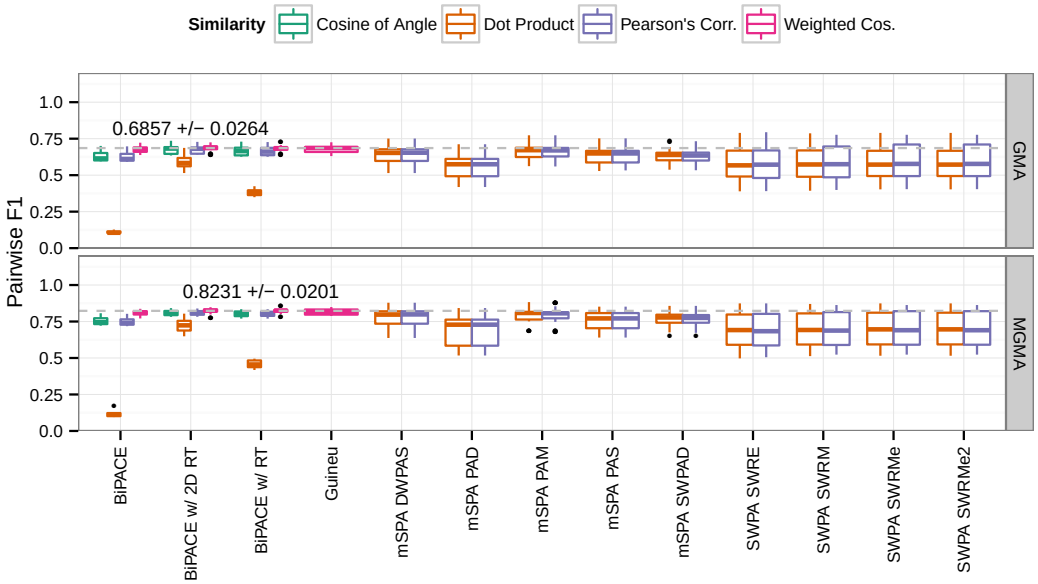


Figure C.13.: Best pairwise average F1 instances for mSPA dataset II. BiPACE 2D with the weighted cosine score achieves the highest average pairwise values and has the lowest standard deviation.

Table C.2.: Best evaluation results for MSPA Dataset II for each algorithm variant concerning achieved F1 score on the GMA and MGMA references. The best value within each column is highlighted in bold face. BIPACE 2D achieves the highest F1 scores. BIPACE has the highest Precision scores for either reference and also has the lowest runtime. GUINEU achieves the highest Recall value.

Reference	Method	F1	Precision	Recall	TP	FP	TN	FN	Unm. in Ref.	Runtime (s)	Memory (MB)
GMA	BiPACE	0.6476	**0.7435**	0.5737	1032	356	310	217	550	**3.36**	116.02
GMA	BiPACE w/ 2D RT	**0.6654**	0.7235	0.6160	1073	410	313	209	460	3.62	114.73
GMA	BiPACE w/ RT	0.6646	0.7218	0.6157	1064	410	327	234	430	3.60	**95.93**
GMA	Guineu	0.6576	0.6492	**0.6663**	**1090**	589	240	**116**	430	3.51	199.51
GMA	mSPA DWPAS	0.6477	0.6770	0.6209	1048	500	277	240	400	138.48	180.80
GMA	mSPA PAD	0.5971	0.6273	0.5697	936	556	266	302	405	165.45	180.40
GMA	mSPA PAM	0.6638	0.6918	0.6379	1073	478	305	229	**380**	159.42	174.00
GMA	mSPA PAS	0.6384	0.6891	0.5945	1022	461	285	237	460	149.73	176.30
GMA	mSPA SWPAD	0.6375	0.6578	0.6183	1019	530	287	234	395	217.93	188.00
GMA	SWPA SWRE	0.6257	0.7323	0.5462	952	348	374	256	535	98.05	211.60
GMA	SWPA SWRM	0.6252	0.7362	0.5432	949	**340**	**378**	263	535	98.03	211.60
GMA	SWPA SWRMe	0.6244	0.7330	0.5438	950	346	372	257	540	100.15	211.60
GMA	SWPA SWRMe2	0.6244	0.7330	0.5438	950	346	372	257	540	101.26	211.60
MGMA	BiPACE	0.7362	**0.7922**	0.6877	808	212	213	102	265	**3.36**	116.02
MGMA	BiPACE w/ 2D RT	**0.7510**	0.7833	0.7213	828	229	223	95	225	3.64	98.17
MGMA	BiPACE w/ RT	0.7510	0.7805	0.7237	825	232	228	100	215	3.99	**98.85**
MGMA	Guineu	0.7390	0.7218	**0.7570**	**838**	323	170	**39**	230	3.58	196.18
MGMA	mSPA DWPAS	0.7350	0.7321	0.7380	817	299	194	105	185	162.23	188.00
MGMA	mSPA PAD	0.6785	0.6800	0.6769	729	343	180	138	210	169.35	174.00
MGMA	mSPA PAM	0.7389	0.7412	0.7366	822	287	197	94	**200**	159.42	174.00
MGMA	mSPA PAS	0.7108	0.7353	0.6879	778	280	189	108	245	149.73	176.30
MGMA	mSPA SWPAD	0.7198	0.7175	0.7221	795	313	186	91	215	187.00	187.70
MGMA	SWPA SWRE	0.7072	0.8087	0.6283	727	172	271	130	300	98.05	211.60
MGMA	SWPA SWRM	0.7050	0.8105	0.6238	723	**169**	**272**	131	305	98.03	211.60
MGMA	SWPA SWRMe	0.7053	0.8101	0.6245	725	170	269	131	305	100.15	211.60
MGMA	SWPA SWRMe2	0.7053	0.8101	0.6245	725	170	269	131	305	101.26	211.60

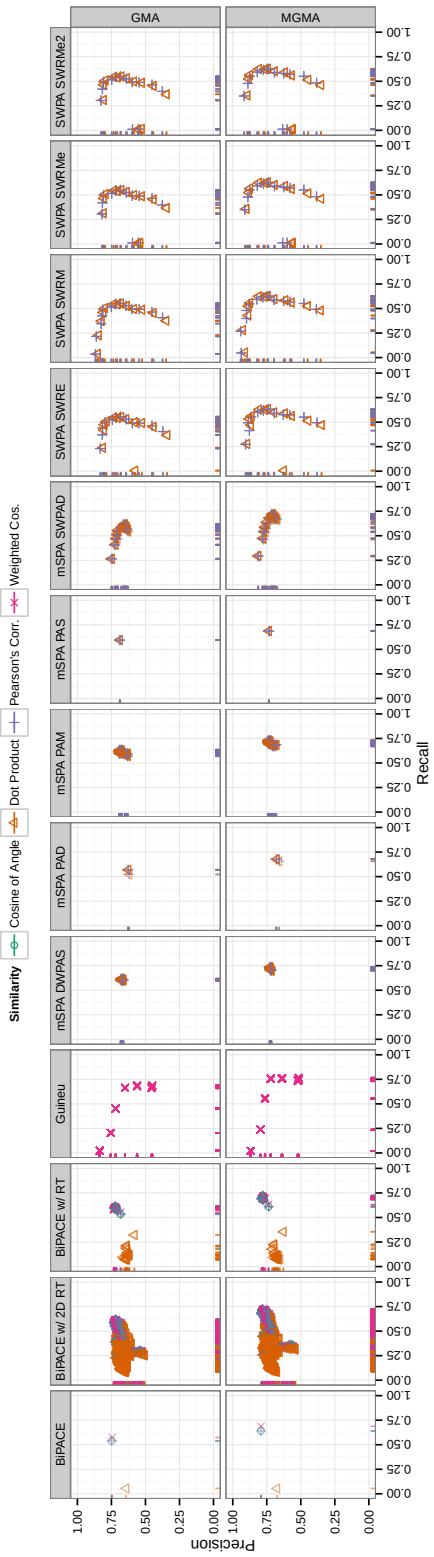


Figure C.14.: Precision and Recall plot for MSPA dataset II. The BIPACE variants consistently achieve Precision values > 0.7 while especially the SWPA variants achieve higher values using the GMA reference. Concerning recall, GUINEU achieves the highest values on both references.

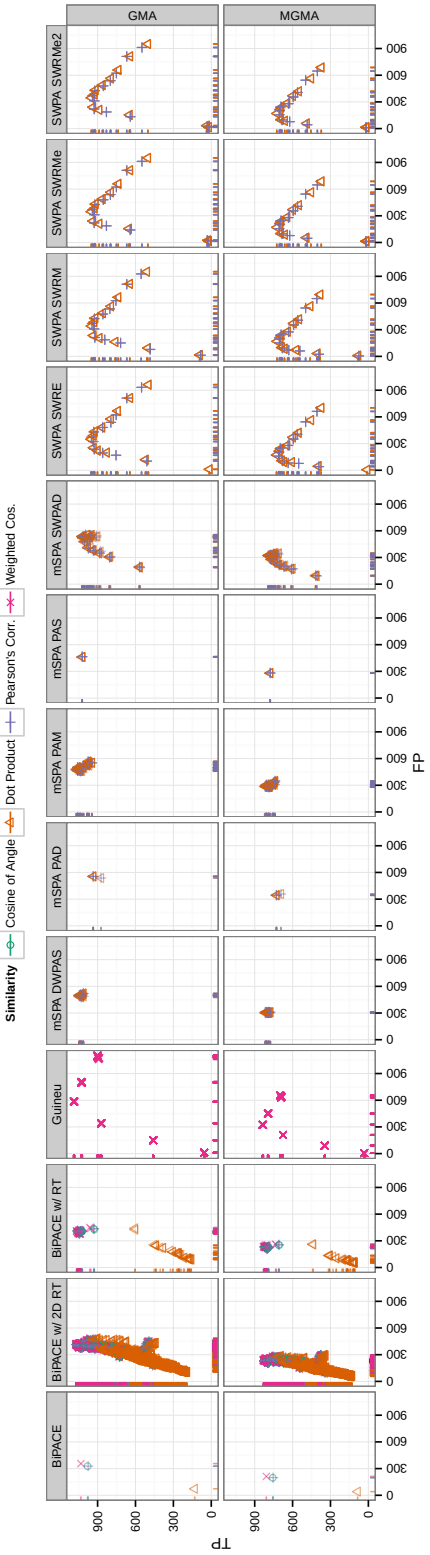


Figure C.15.: False Positives vs. True Positives for MSPA dataset II. Here, GUINEU achieves the highest number of TPs on either reference. The number of FPs is comparable or lower than the number of FPs achieved by either MSPA or SWPA variant, for a comparable number of TPs.

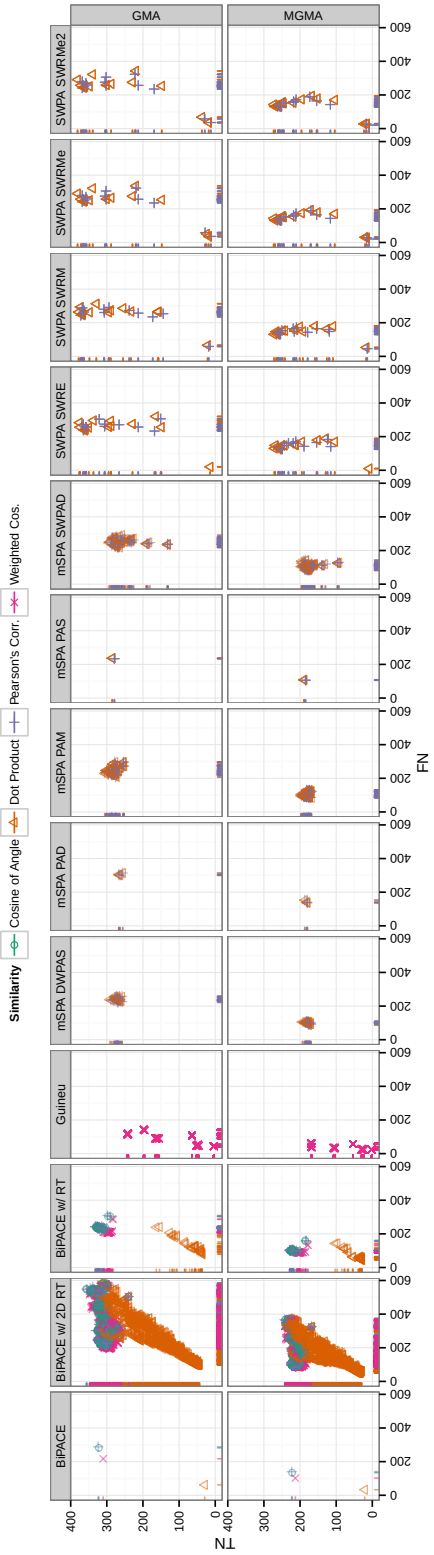


Figure C.16.: False Negatives vs. True Negatives for MSPA dataset II. Most of the BIPACE variants achieve considerably higher TN values than either MSPA or SWPA variant. However, it is visible that the BIPACE variants tend to have a much larger number of FNs, due to the more conservative peak grouping approach, which leads to a higher number of peaks reported as missing in the final alignment. GUINEU and the MSPA variants achieve the lowest FN values.

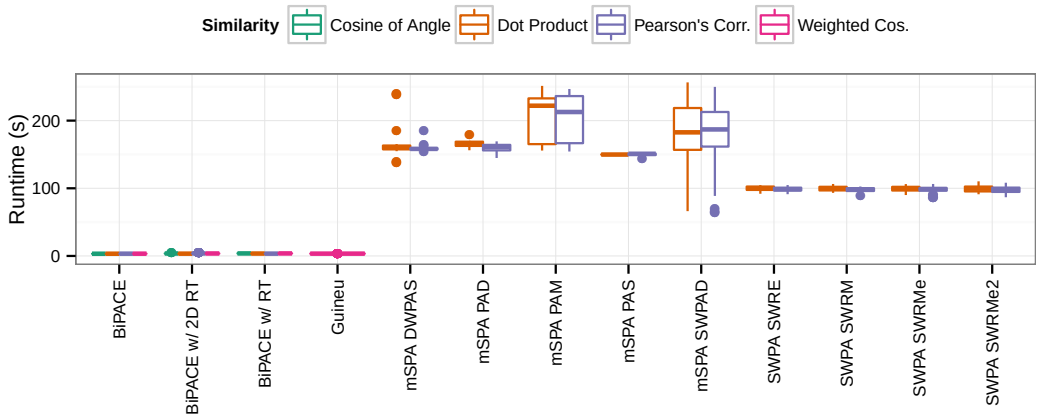


Figure C.17.: Runtime plot for MSPA dataset II. The BIPACE variants and GUINEU all have runtimes below five seconds. The SWPA variants run about 100 seconds, while most of the MSPA variants have runtimes exceeding 150 seconds.

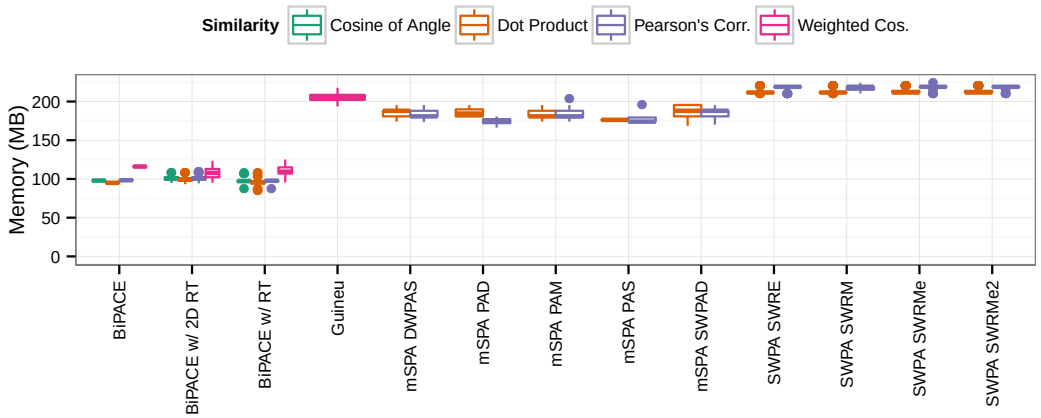


Figure C.18.: Memory plot for MSPA dataset II. The MSPA, SWPA, and GUINEU variants all use about 180-200 MBytes of memory, while most BIPACE instances using the cosine, dot product, and Pearson's linear correlation as mass spectral similarities use around 100-120 MBytes of memory.

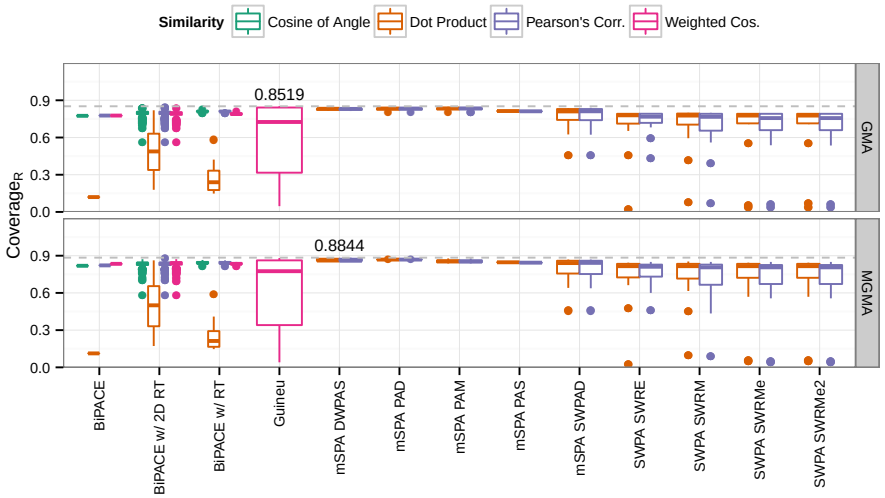


Figure C.19.: Coverage$_R$ plot for MSPA dataset II. The best BIPACE, GUINEU, and MSPA instances achieve a coverage of around 85% concerning the peaks contained in the reference alignment, while SWPA achieves lower numbers.

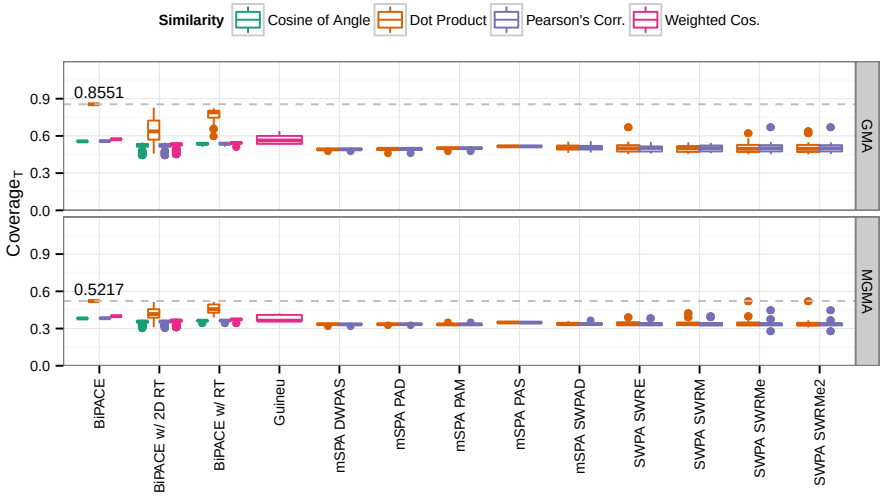


Figure C.20.: Coverage$_T$ plot for MSPA dataset II. Here, the best BIPACE variants using the dot product can assign the largest number (85%, 52%) of their aligned peaks to the reference alignments, while the other methods have considerably lower numbers.

C.4. SWPA Dataset I Evaluation Results

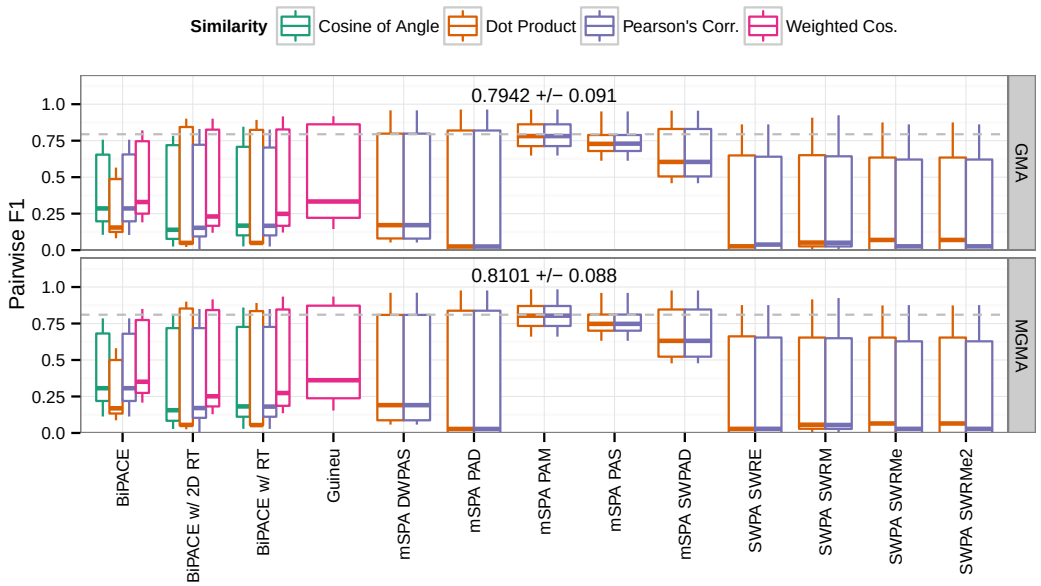


Figure C.21.: Best pairwise average F1 instances for SWPA dataset I. mSPA-PAM achieves the highest average pairwise values and has the lowest standard deviation. All other methods show a considerably large standard deviation and rather low median values for the pairwise F1 score.

Table C.3.: Best evaluation results for SWPA Dataset I for each algorithm variant concerning achieved F1 score on the GMA and MGMA references. The best value within each column is highlighted in bold face. MSPA-PAM achieves the highest F1, Precision and Recall scores. GUINEU has the lowest runtime and the BIPACE variants consume the least amount of memory.

Reference	Method	F1	Precision	Recall	TP	FP	TN	FN	Unm. in Ref.	Runtime (s)	Memory (MB)
GMA	BiPACE	0.6584	0.9049	0.5174	609	64	87	424	144	3.39	93.34
GMA	BiPACE w/ 2D RT	0.6953	0.8588	0.5841	663	109	84	424	48	4.96	92.41
GMA	BiPACE w/ RT	0.6966	0.8801	0.5764	668	91	78	427	64	3.97	**91.06**
GMA	Guineu	0.7061	0.8163	0.6221	693	156	58	309	112	**3.15**	131.87
GMA	mSPA DWPAS	0.6732	0.8408	0.5613	618	117	**110**	483	**0**	38.27	116.20
GMA	mSPA PAD	0.6656	0.8413	0.5506	615	116	95	486	16	35.70	116.20
GMA	mSPA PAM	**0.8366**	**0.9397**	**0.7538**	**888**	**57**	93	**242**	48	41.16	116.20
GMA	mSPA PAS	0.7994	0.9206	0.7064	823	71	92	310	32	31.23	116.20
GMA	mSPA SWPAD	0.7657	0.9087	0.6616	776	78	77	333	64	30.18	116.20
GMA	SWPA SWRE	0.5212	0.8315	0.3796	454	92	40	582	160	18.43	115.40
GMA	SWPA SWRM	0.5220	0.7904	0.3897	445	118	68	569	128	18.81	115.40
GMA	SWPA SWRMe	0.5178	0.8138	0.3797	437	100	77	618	96	19.21	115.40
GMA	SWPA SWRMe2	0.5187	0.8172	0.3799	438	98	77	619	96	19.43	115.40
MGMA	BiPACE	0.6788	0.9146	0.5397	578	54	75	365	128	3.39	93.34
MGMA	BiPACE w/ 2D RT	0.7127	0.8631	0.6070	624	99	73	372	32	4.96	92.41
MGMA	BiPACE w/ RT	0.7136	0.8834	0.5985	629	83	66	374	48	3.97	**91.06**
MGMA	Guineu	0.7148	0.8181	0.6347	643	143	44	258	112	**3.15**	131.87
MGMA	mSPA DWPAS	0.6814	0.8360	0.5750	571	112	**95**	422	**0**	36.44	116.20
MGMA	mSPA PAD	0.6785	0.8519	0.5637	575	100	80	429	16	35.70	116.20
MGMA	mSPA PAM	**0.8517**	**0.9475**	**0.7735**	**830**	**46**	81	**211**	32	41.16	116.20
MGMA	mSPA PAS	0.8129	0.9310	0.7214	769	57	77	265	32	31.23	116.20
MGMA	mSPA SWPAD	0.7807	0.9169	0.6797	728	66	63	279	64	30.18	116.20
MGMA	SWPA SWRE	0.5301	0.8294	0.3895	423	87	27	519	144	18.43	115.40
MGMA	SWPA SWRM	0.5194	0.7791	0.3895	402	114	54	518	112	18.23	115.40
MGMA	SWPA SWRMe	0.5185	0.8093	0.3815	399	94	60	567	80	19.21	115.40
MGMA	SWPA SWRMe2	0.5195	0.8130	0.3817	400	92	60	568	80	19.43	115.40

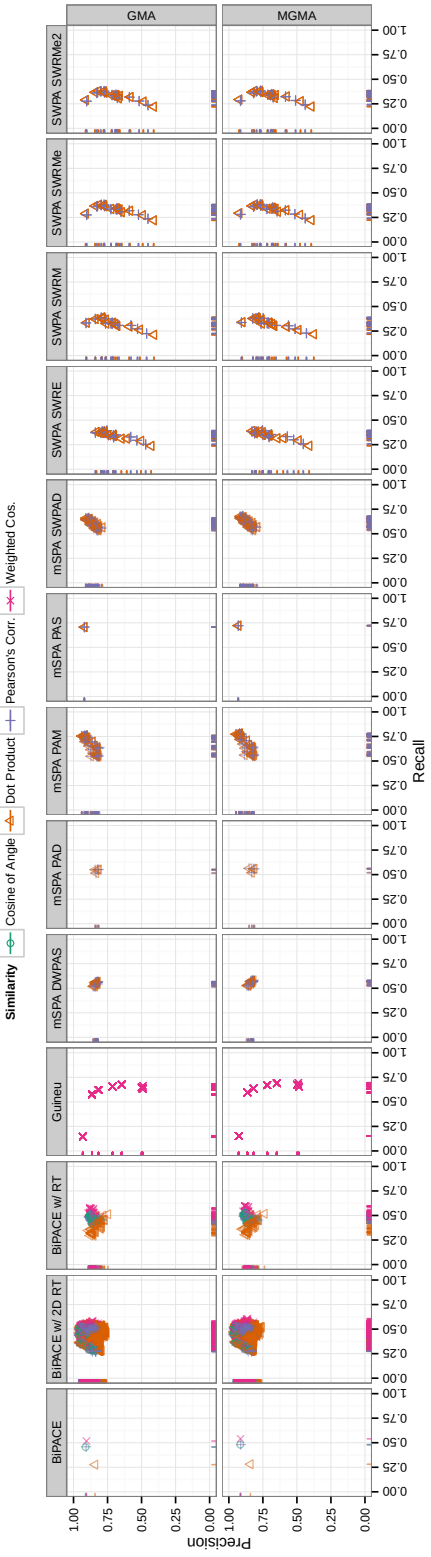


Figure C.22.: Precision and Recall plot for SWPA dataset I. The mSPA-PAM instances achieve the highest Recall value for either reference alignment, while the other mSPA instances are not far behind. Concerning Precision values, the BiPACE 2D instances are close to mSPA-PAM, but lack in Recall. SWPA and its variants achieve a good Precision value, fall behind in Recall.

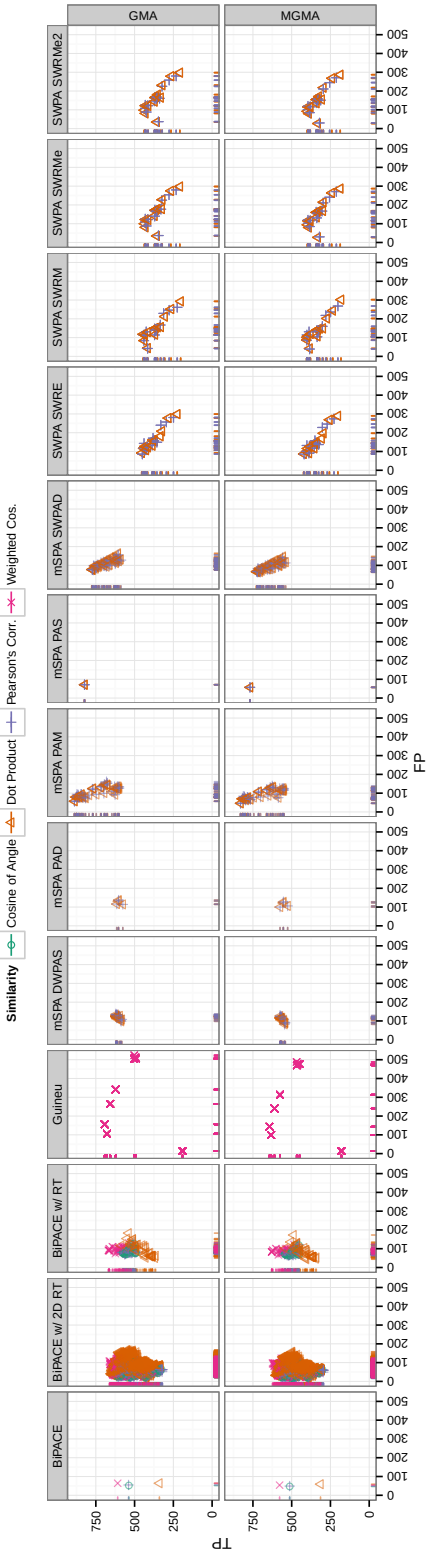


Figure C.23.: False Positives vs. True Positives for SWPA dataset I. mSPA-PAM achieves the highest TP values, as well as the lowest FP values, on either reference, followed by mSPA-PAS and mSPA-SW-PAD.

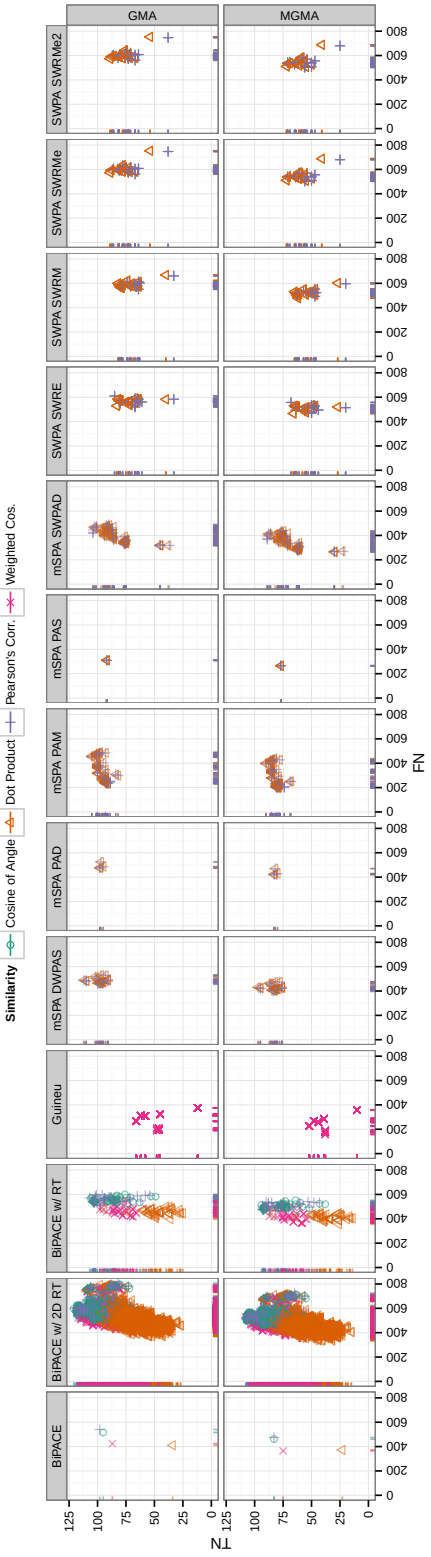


Figure C.24.: False Negatives vs. True Negatives for SWPA dataset I. The BiPACE variants tend to achieve very high FN numbers hinting at the fact that the alignments lacked many peaks that were reported by the other methods. Nonetheless, BiPACE and its variants achieve considerably higher TN values than either mSPA or SWPA variant, thus they recover more peaks marked as missing in the reference alignment.

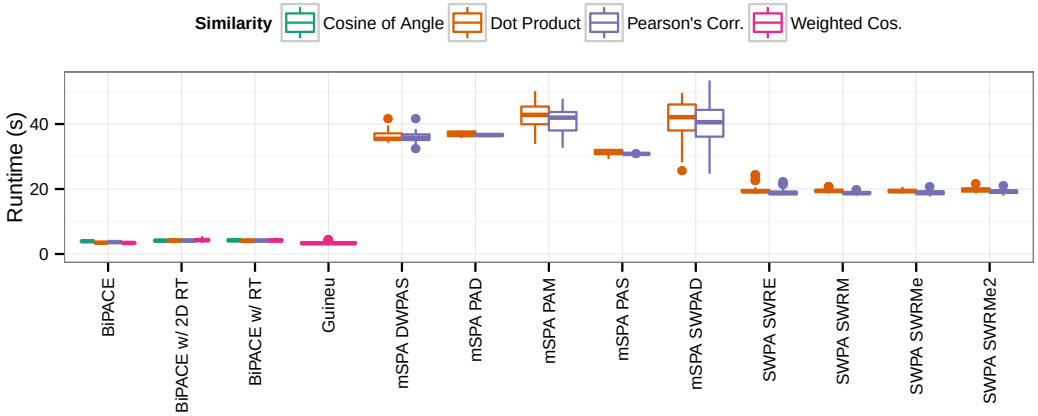


Figure C.25.: Runtime plot for SWPA dataset I. The runtime of the BIPACE instances varies between 3 and 4 seconds, while GUINEU runs in close to 3 seconds. The MSPA variants all run for at least 30 seconds, some even for more than 45 seconds. SWPA variants vary in runtime around 20 seconds.

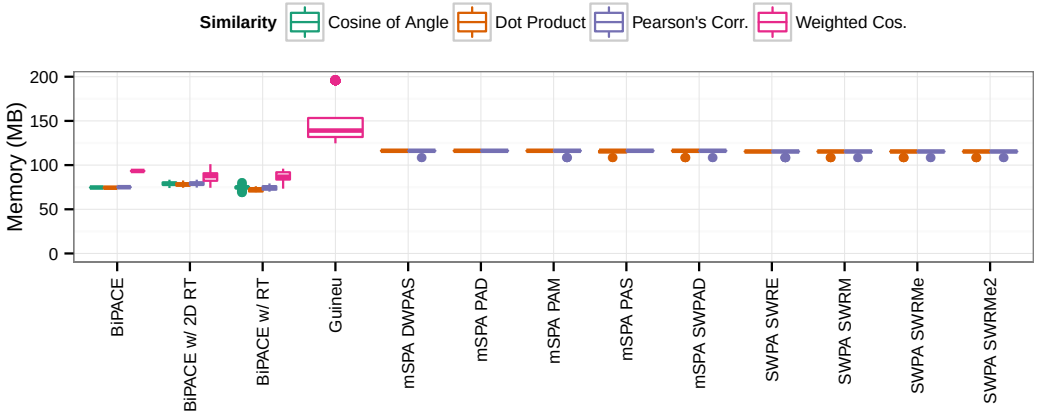


Figure C.26.: Memory plot for SWPA dataset I. The SWPA and MSPA variants consume around 115 MBytes of main memory. The BIPACE instances show a larger variation, with a majority consuming between 70 and 80 MBytes. Some instances require larger amounts of memory due to more relaxed parameters and due to the mass spectral similarity used.

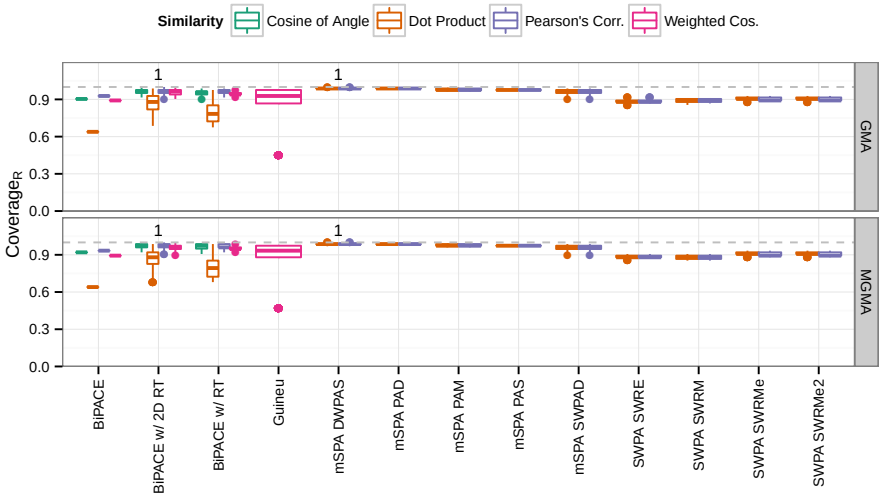


Figure C.27.: Coverage$_R$ plot for SWPA dataset I. The reference coverage for the BIPACE and MSPA variants varies between 90% and almost 100%, while SWPA covers at most slightly more than 90% of the peaks contained in the reference. Thus, the MSPA and BIPACE variants achieve a considerably larger coverage of the reference alignment.

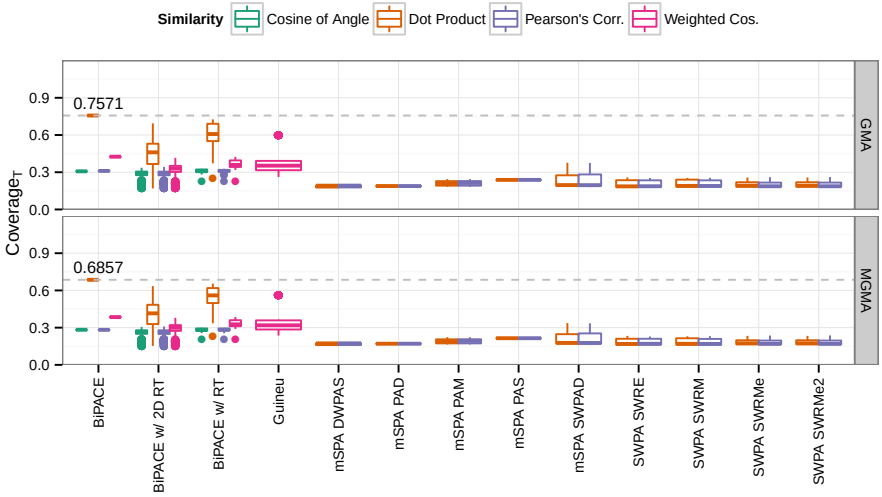


Figure C.28.: Coverage$_T$ plot for SWPA dataset I. Here, the tool coverage shows that most methods report many more aligned peak groups than are contained in the reference alignment. A high value, like in the case of BIPACE using the dot product similarity indicates, that not many peak groups were actually found, but those that were, could be assigned to 75% and 68% of reference peak groups, respectively.

C.5. CHLAMY Dataset I Evaluation Results

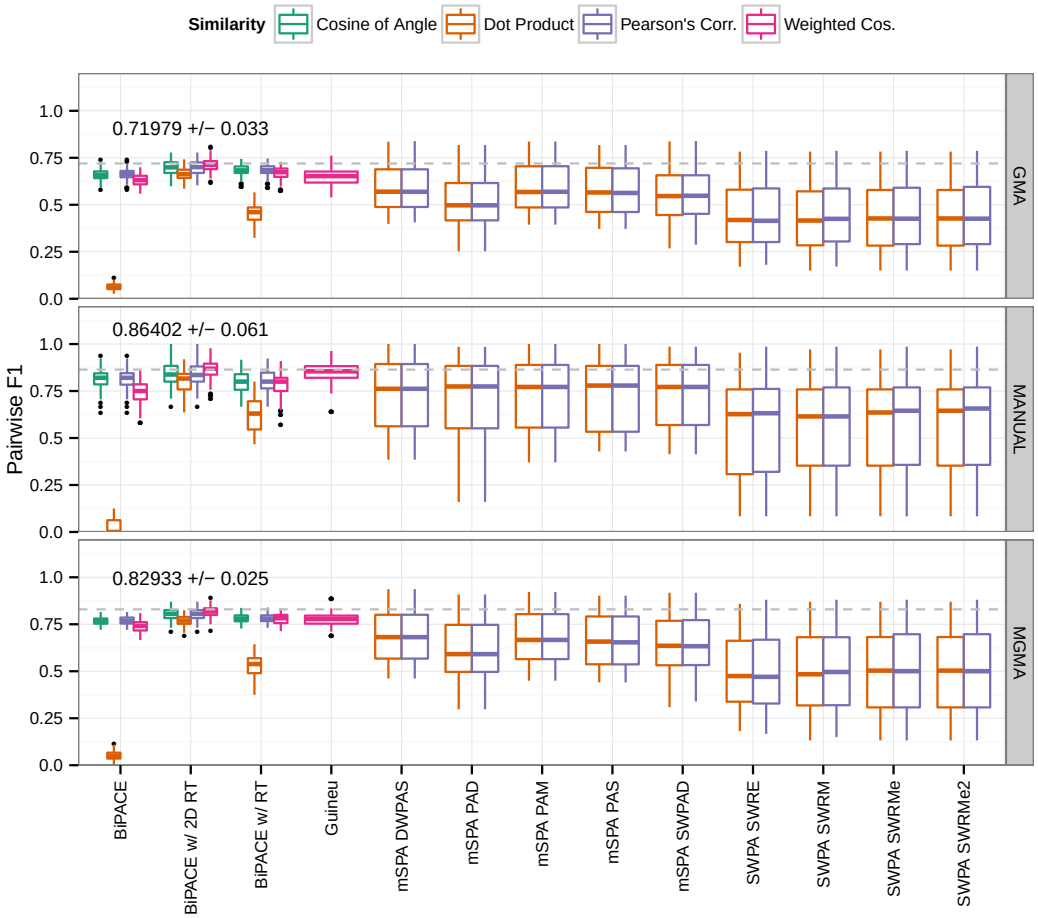


Figure C.29.: Best pairwise average F1 instances for CHLAMY dataset I. BiPACE 2D with the weighted cosine score achieves the highest average pairwise values and has the lowest standard deviation.

Table C.4.: Best evaluation results for CHLAMY Dataset I for each algorithm variant concerning achieved F1 score on the GMA, MANUAL, and MGMA references. The best value within each column is highlighted in bold face. BIPACE 2D achieves the highest F1 and Recall scores, while BIPACE gains the highest Precision.

Reference	Method	F1	Precision	Recall	TP	FP	TN	FN	Unm. in Ref.	Runtime (s)	Memory (MB)
GMA	BiPACE	0.6507	**0.8466**	0.5284	1612	**292**	1085	731	708	**3.92**	110.97
GMA	BiPACE w/ 2D RT	**0.6692**	0.7998	**0.5752**	**1702**	426	1043	585	672	4.78	**107.59**
GMA	BiPACE w/ RT	0.6482	0.8086	0.5408	1622	384	1045	633	744	4.32	111.41
GMA	Guineu	0.5401	0.6844	0.4461	1390	641	671	**370**	1356	5.00	206.37
GMA	mSPA DWPAS	0.5923	0.7668	0.4824	1401	426	1098	939	**564**	190.89	157.90
GMA	mSPA PAD	0.5693	0.7412	0.4621	1346	470	1045	979	588	130.35	163.80
GMA	mSPA PAM	0.6027	0.7821	0.4902	1425	397	1124	894	588	126.54	157.90
GMA	mSPA PAS	0.5842	0.7736	0.4693	1377	403	1091	921	636	127.43	157.90
GMA	mSPA SWPAD	0.5894	0.7687	0.4780	1379	415	**1128**	942	**564**	178.34	157.90
GMA	SWPA SWRE	0.4803	0.7888	0.3453	1128	302	859	1131	1008	83.74	169.90
GMA	SWPA SWRM	0.4895	0.7600	0.3610	1140	360	910	1130	888	81.67	176.40
GMA	SWPA SWRMe	0.4816	0.7602	0.3524	1116	352	909	1175	876	79.38	169.90
GMA	SWPA SWRMe2	0.4823	0.7590	0.3535	1118	355	910	1169	876	84.00	169.90
MANUAL	BiPACE	0.7310	**0.7539**	0.7095	337	**110**	231	78	60	**3.92**	110.97
MANUAL	BiPACE w/ 2D RT	**0.7430**	0.7271	**0.7596**	**357**	134	212	53	60	4.78	**107.59**
MANUAL	BiPACE w/ RT	0.7320	0.7320	0.7320	336	123	234	87	36	4.41	108.72
MANUAL	Guineu	0.6892	0.7874	0.6128	326	88	196	**62**	144	5.15	204.45
MANUAL	mSPA DWPAS	0.7239	0.7573	0.6933	312	100	266	102	**36**	133.46	163.80
MANUAL	mSPA PAD	0.7138	0.7400	0.6894	313	110	252	105	36	131.40	163.80
MANUAL	mSPA PAM	0.7277	0.7441	0.7120	314	108	267	103	24	129.03	157.90
MANUAL	mSPA PAS	0.7067	0.7574	0.6623	306	98	256	108	48	132.34	157.90
MANUAL	mSPA SWPAD	0.7262	0.7542	0.7002	313	102	**267**	110	**24**	163.42	157.90
MANUAL	SWPA SWRE	0.6221	0.8612	0.4869	242	39	280	171	84	82.34	176.40
MANUAL	SWPA SWRM	0.6334	0.7449	0.5510	254	87	268	147	60	81.43	169.90
MANUAL	SWPA SWRMe	0.6236	0.7404	0.5386	251	88	262	155	60	79.38	169.90
MANUAL	SWPA SWRMe2	0.6236	0.7404	0.5386	251	88	262	155	60	84.00	169.90

Continued on next page ↷

Table C.4 – continued from previous page

Reference	Method	F1	Precision	Recall	TP	FP	TN	FN	Unm. in Ref.	Runtime (s)	Memory (MB)
MGMA	BiPACE	0.7389	**0.8844**	0.6345	1132	**148**	756	340	312	**4.68**	108.89
MGMA	BiPACE w/ 2D RT	**0.7662**	0.8349	**0.7079**	**1229**	243	709	243	264	4.78	**107.59**
MGMA	BiPACE w/ RT	0.7359	0.8420	0.6536	1151	216	711	298	312	4.11	110.29
MGMA	Guineu	0.6556	0.7418	0.5873	1086	378	461	**175**	588	5.00	206.37
MGMA	mSPA DWPAS	0.6757	0.7759	0.5984	997	288	734	477	**192**	135.68	163.80
MGMA	mSPA PAD	0.6328	0.7648	0.5396	940	289	657	514	288	130.35	163.80
MGMA	mSPA PAM	0.6735	0.8171	0.5728	983	220	752	505	228	126.54	157.90
MGMA	mSPA PAS	0.6502	0.8057	0.5449	958	231	699	476	324	132.34	157.90
MGMA	mSPA SWPAD	0.6613	0.8076	0.5599	982	234	**700**	496	**276**	172.49	163.80
MGMA	SWPA SWRE	0.5474	0.7861	0.4199	768	209	650	665	396	81.48	176.30
MGMA	SWPA SWRM	0.5551	0.7947	0.4265	778	201	663	662	384	81.43	169.90
MGMA	SWPA SWRMe	0.5495	0.8195	0.4133	772	170	650	652	444	82.57	169.90
MGMA	SWPA SWRMe2	0.5510	0.8184	0.4153	775	172	650	647	444	80.67	169.90

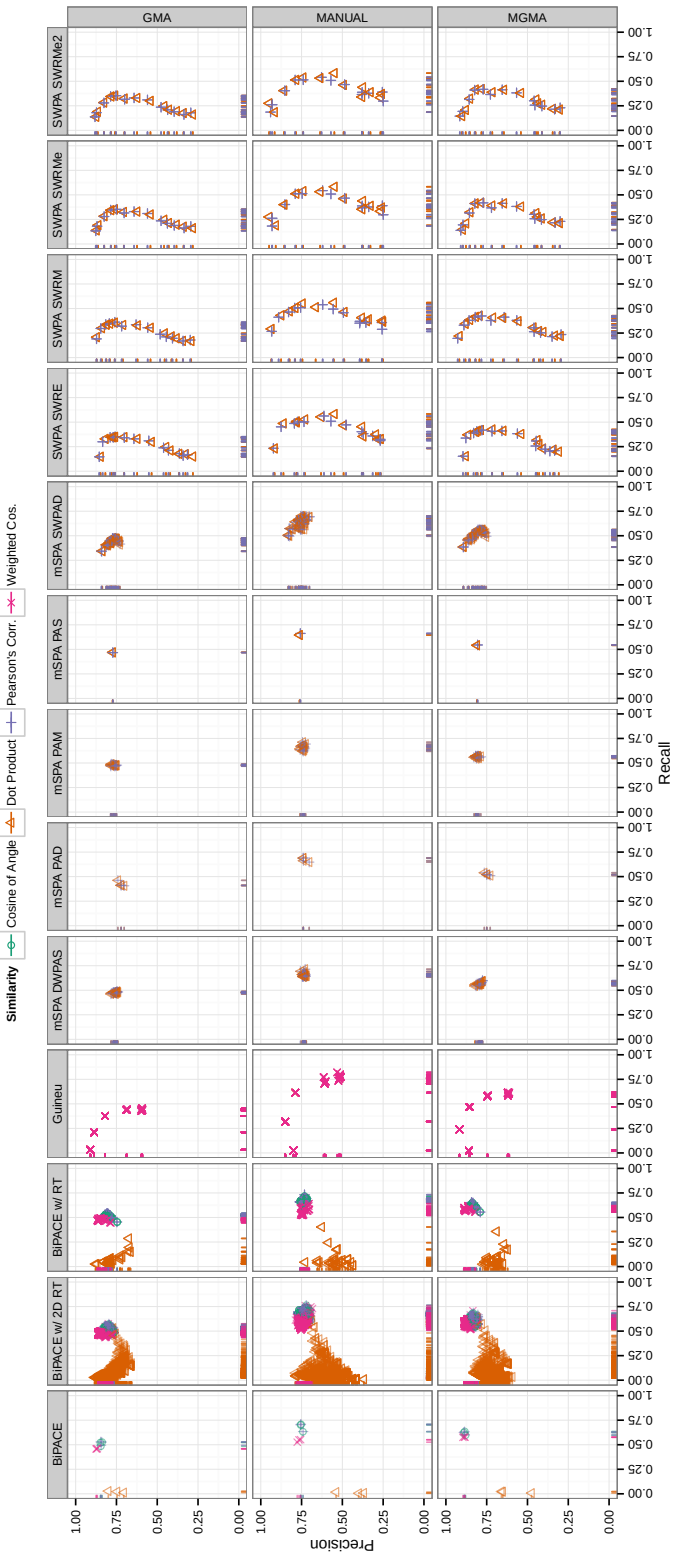


Figure C.30.: Precision and Recall plot for CHLAMY dataset I. For the GMA reference alignment, the SWPA methods and GUINEU achieve the highest Precision values, but at the cost of very low Recall values around 0.25. For the MANUAL and MGMA references, BIPACE 2D achieves the highest Recall values. The MSPA variants have relatively low Recall values (> 0.5 and < 0.75), while their Precision values range between 0.7 and 0.79.

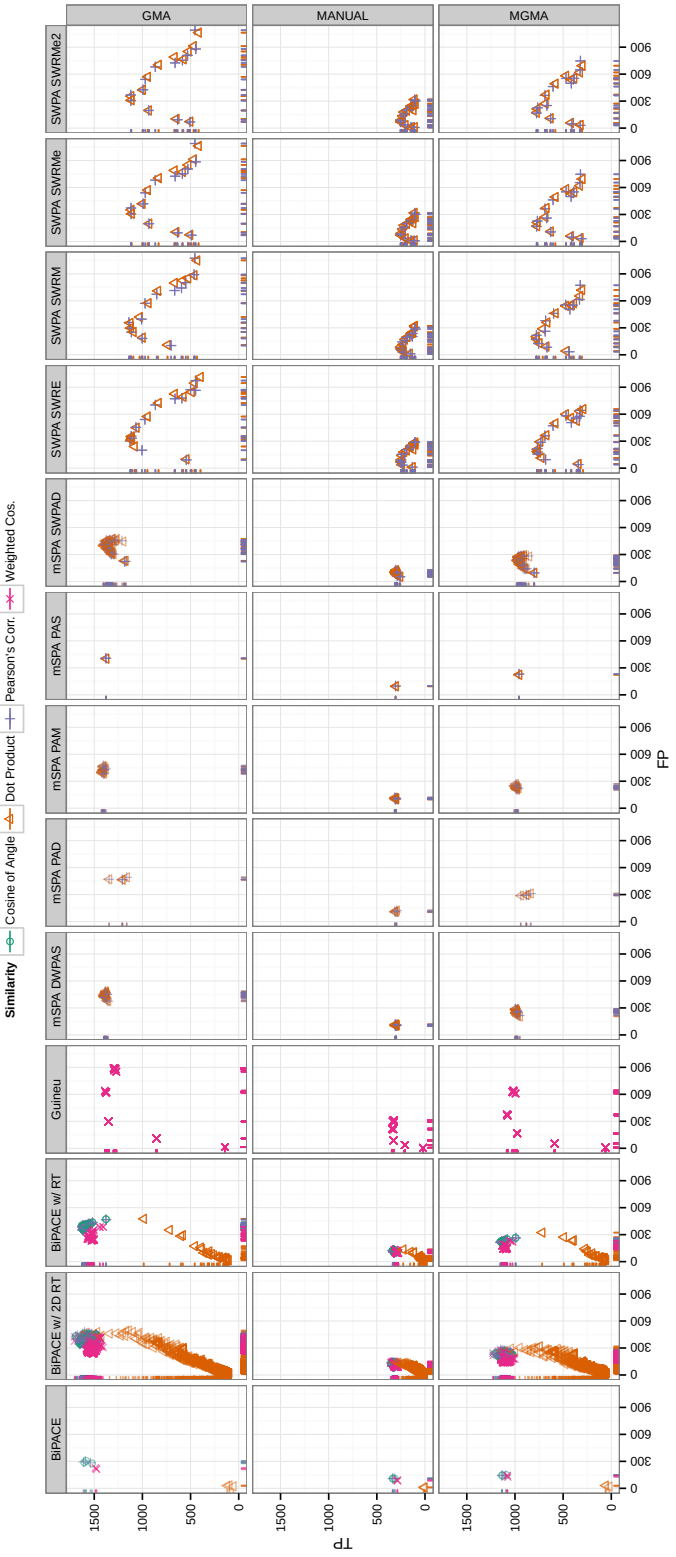


Figure C.31.: False Positives vs. True Positives for CHLAMY dataset I. In absolute TP numbers, BIPACE and its variants perform best on all three reference alignments. GUINEU achieves a high number of TPs as well, but also has high FP values.

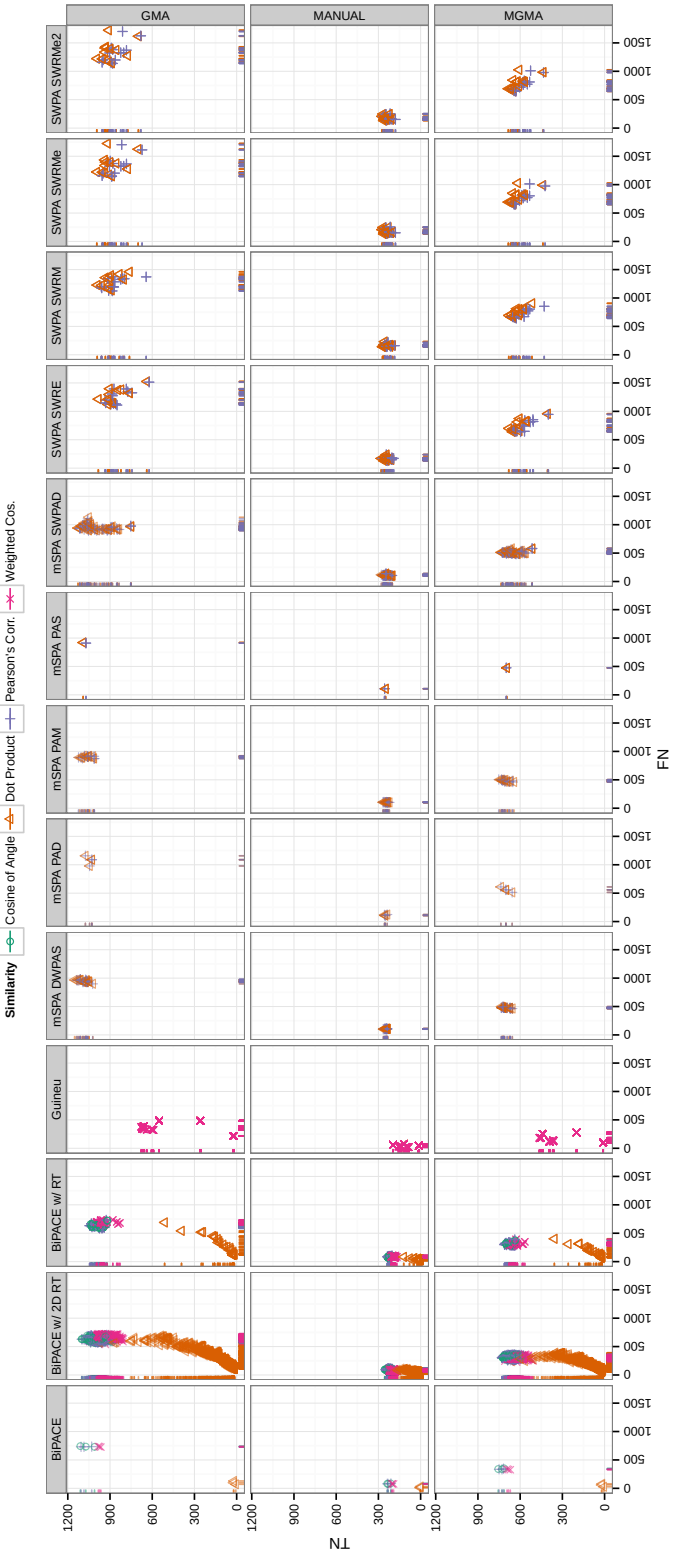


Figure C.32.: False Negatives vs. True Negatives for CHLAMY dataset I. The best BIPACE 2D and BIPACE RT instances have considerably larger values for TNs as well as lower FN values than any MSPA or SWPA variant. Thus, the BIPACE variants miss fewer peaks present in the reference alignments but also correctly report peaks that are really missing in the reference.

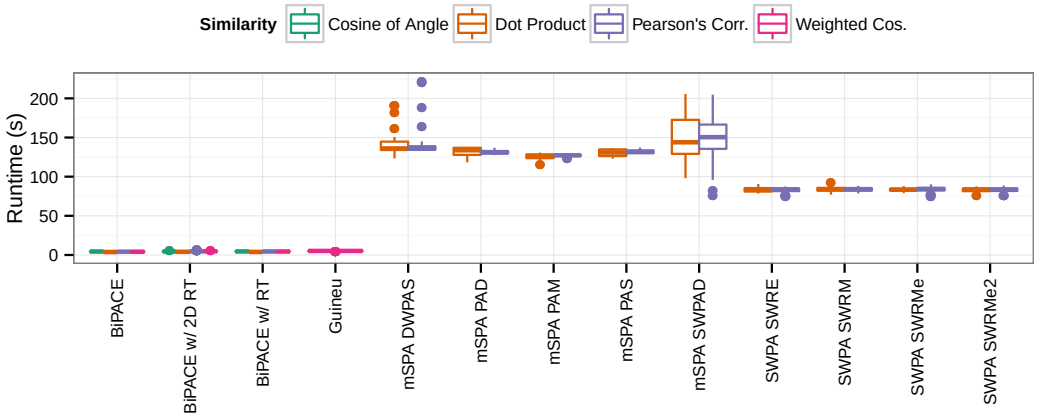


Figure C.33.: Runtime plot for CHLAMY dataset I. The runtimes for the BiPACE variants are below 5 seconds, while the SWPA instances run for around 85 seconds. The mSPA variants show the largest runtime variability between 120 to around 140 seconds. The GUINEU instances have a runtime of about 5 seconds.

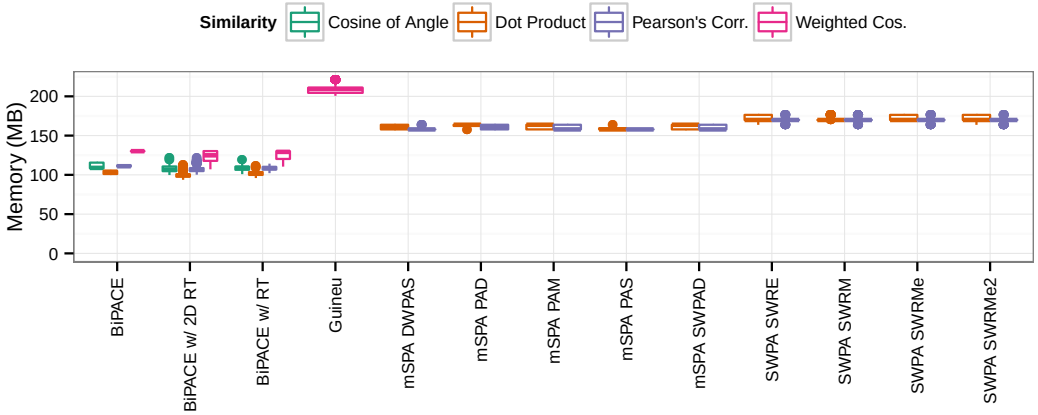


Figure C.34.: Memory plot for CHLAMY dataset I. The mSPA and SWPA instances consume between 160 and 170 MBytes of memory while BiPACE and its variants consume between 100 and 130 MBytes of memory. GUINEU consumes around 200 MBytes of memory.

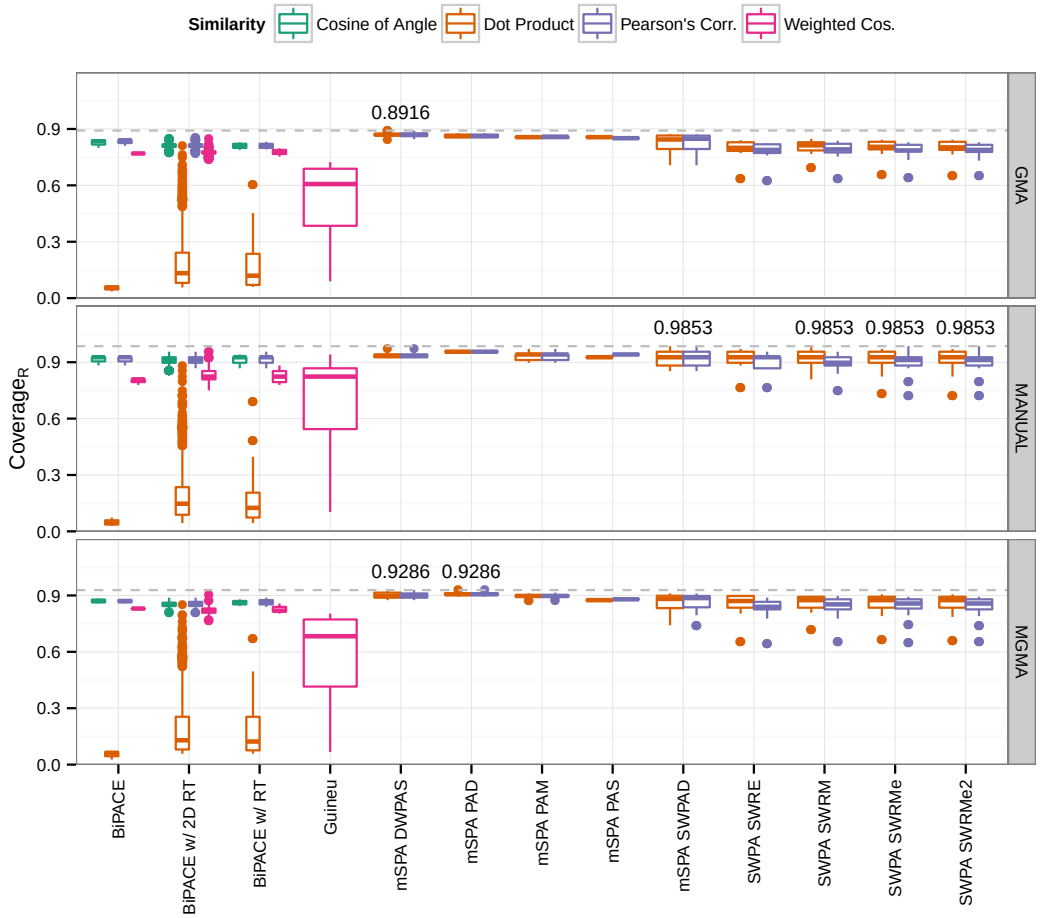


Figure C.35.: Coverage$_R$ plot for CHLAMY dataset I. The reference coverage for all algorithms, except for GUINEU, on the different references is between 70 and 90%. Due to the reduced size of the MGMA alignment reference in comparison to the GMA reference alignment, a higher coverage is expected. The manual reference shows higher coverage values for all algorithms.

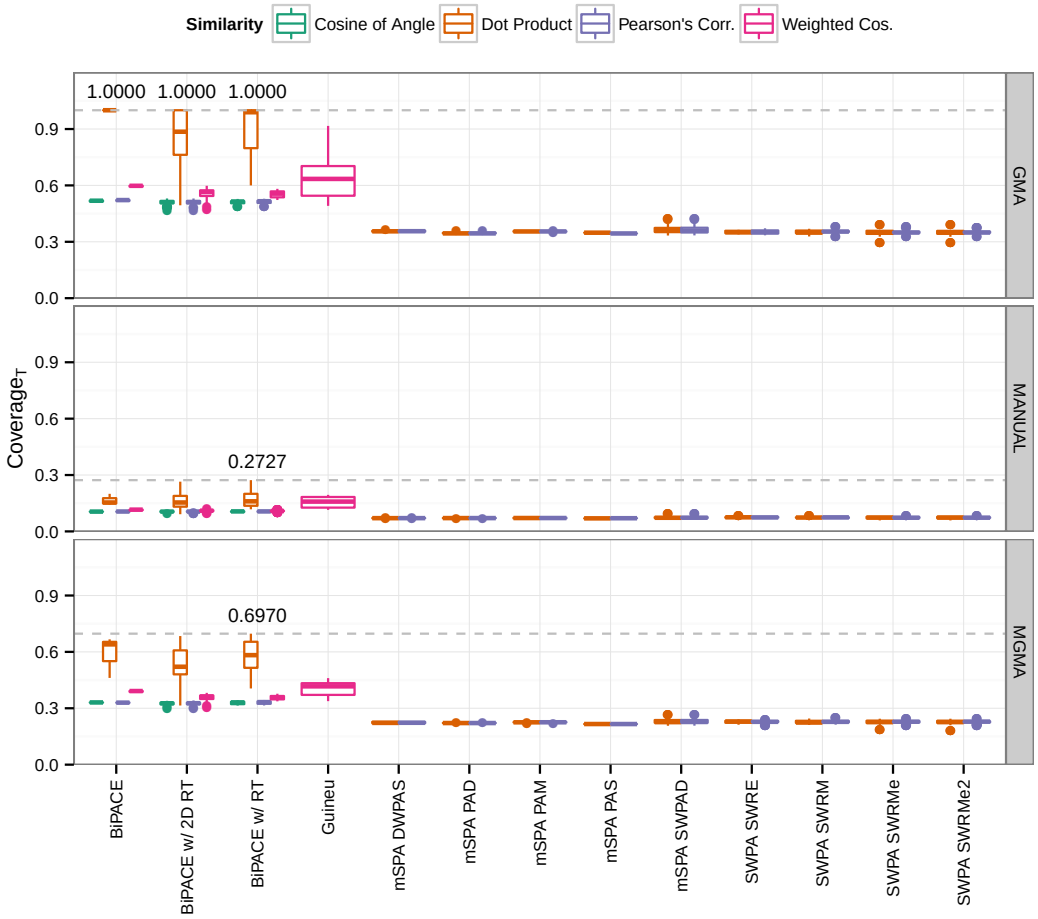


Figure C.36.: Coverage$_T$ plot for CHLAMY dataset I. The coverage of the reported tool peak alignments varies largely between GMA and MGMA variants. This is a result of the exclusion of a larger number of peaks failing the consistency criterion, which may exclude valid peaks. Finally, due to its constrained size, the tool alignment coverage for the MANUAL reference alignment is lower, since most algorithm instances will report many more aligned peak groups, than are present in the reference alignment.

C.6. Parameter Selection for BIPACE 2D

C.6.1. Parameters influencing alignment quality

Good starting parameters for BIPACE 2D are $D1 = \hat{\sigma}(t_1)$, $D2 = \hat{\sigma}(t_2)$, effectively the expected standard deviation in the first and second retention time, and $T1 = 0.0$ and $T2 = 0.0$, as the retention time matching thresholds to control the number of false positives. Peaks with peak retention time deviations resulting in a function value of the Gaussian retention penalty term (function value ranges between 0 and 1) below the given threshold ($T1$ or $T2$) are removed from further consideration.

The minimum clique size parameter only influences the output of BIPACE 2D, so it is a simple filter to select only cliques that match or exceed the given parameter value. As the similarity function, the weighted cosine has proven to be both fast and precise.

C.6.2. Setting the parameters for BIPACE 2D

BIPACE 2D is included in the MALTCMS software framework that is available from http://maltcms.sf.net. After downloading and extracting the MALTCMS distribution, the directory structure contains a folder termed cfg. Below that folder, the pipelines/xml folder contains the definition of various pipelines that can be executed with MALTCMS. The folder pipelines contains one .mpl properties file for each XML-based pipeline definition in the xml folder. The configuration file bipace2D.xml contains the configuration information for the parameters used by BIPACE 2D.

These include, as direct parameters of the bean tag element with id peakCliqueAlignment:

- *MCS* =minCliqueSize - the minimum clique size to be reported in the multiple alignment
- saveUnmatchedPeaks - whether peaks without a best hit should be saved to an MSP compatible file
- saveUnassignedPeaks - whether peaks that are not a part of a biclique or bidirectional best hit should be saved to an MSP compatible file
- saveIncompatiblePeaks - whether peaks that were not mergeable into larger cliques should be saved to an MSP compatible file

Other parameters are configured in other bean tags, that are referenced by their id in the peakCliqueAlignment tag.

Array Similarity (referenced in bean with id timePenalizedProductSimilarity), one of:

- dotSimilarity - dot product similarity
- cosineSimilarity - cosine similarity
- linCorrSimilarity - Pearson's linear correlation coefficient

- weightedCosineSimilarity - weighted cosine similarity

Retention Time Penalties (referenced in bean with id timePenalizedProductSimilarity):

- first rt dimension: bean with id gaussianDifferenceSimilarityRt1:
  - $D1$ =tolerance - the retention time tolerance in the first separation dimension
  - $T1$ =threshold - the retention time tolerance threshold in the first separation dimension
- second rt dimension: bean with id gaussianDifferenceSimilarityRt2:
  - $D2$ =tolerance - the retention time tolerance in the second separation dimension
  - $T2$ =threshold - the retention time tolerance threshold in the second separation dimension

Maximum search range for first and second column retention times (in bean with id worker2DFactory):

- maxRTDifferenceRt1 - maximum retention time difference on first column to include in comparison, usually $= 2 * D1$
- maxRTDifferenceRt2 - maximum retention time difference on second column to include in comparison, usually $= 2 * D2$

C.6.3. Converting ChromaTOF peak lists to netCDF format

- Download the latest release of the MALTCMS User Interface Maui (http://maltcms.sf.net/maui)
- Install the application and create a new project
- Select the ChromaTOF peak reports as your 'Data Files'
- Assign groups to your peak reports
- Set the 'Separation Type' to GCxGC
- Set the 'Detector Type' to TOF-MS (for Leco Pegasus IV data)
- Select 'Override' and set 'Modulation Time' and 'Scan Rate' according to your setup
- Finish the wizard and wait for the import to complete
- The converted files are now below the project's directory, below import/ChromaTofPeakListImporter and a time stamp indicating the date and time of the import.

C.6.4. Running BIPACE 2D

Change to the directory where you extracted the downloaded MALTCMS version and run:

```
> bin/maltcms.sh -f path/to/converted/files/*.cdf -c cfg/pipelines/chroma2D.mpl
```

The program will start up and print its progress to the terminal. When it is finished, it will have created a folder maltcmsOutput below which the results are stored.

C.7. Discussion of Pairwise Alignment vs. Row Wise Multiple Alignment Evaluation

In order to compare the multiple alignments generated by the algorithms under consideration in this manuscript, a quality measure that captures how well the algorithms have reproduced the given reference multiple alignment has to be defined. In principle, there are two possible ways to approach this.

The first approach evaluates the performance of all pairwise alignments individually against the corresponding columns of the reference multiple alignment, as used by Kim, Fang, et al. 2011. However, this approach can not count double or larger gaps, spanning multiple peak lists (columns in the multiple alignment). This would be necessary in order to assess the number of true negatives in relation to the reference alignment. Kim, Fang, et al. 2011 count the TP, FP, FN, and TN numbers in the following way: They count, based on the comparison of pairwise alignments, how many of the pairs between reference and result have the same identity (in this case, they use the name as identity criterion) (TP), a different identity (FP), were not aligned, but should have (FN), and those that were correctly not aligned (TN). The last number is defined as the product of the length of the reference peak list m multiplied with the length of the reported peak list n minus the length of the reported positive peaks u minus the FPs ($mn - u - FP$).

The second approach tries to achieve a related but slightly different goal by defining the reference multiple alignment as the ground truth against which the aligned peak rows reported by the different algorithms are tested in turn. In that sense, we try to cover each row of the reference alignment with the best matching row as reported by the peak alignment algorithm. Then, we count for each matching peak pair between reference and result a TP, a FP when the result reports a peak, whereas the reference did not contain a peak at that position. We count an FN if the result did not report a peak at that position, whereas the reference did and finally, we count peaks that are reported as absent in the result and the reference as TN. Additionally, all non-missing peaks within reference groups that have not been covered by any result group are counted as additional FN (unassigned peaks), to normalize for the total number of peaks contained within the reference alignment. Thus, our $F1$ score is automatically normalized towards the number of peaks contained in the reference alignment.

Thus, in the second approach, TP+FP+FN+TN+UNASSIGNED = number of Peaks in Ground Truth. This is also supported by our $Coverage_T$ (Coverage of Peaks reported by Tool against the reference) and $Coverage_R$ (Coverage of Peaks within the reference covered by the peaks reported by the tool, both measures shown in this section) measures. These also show, that both MSPA and SWPA achieve a good $Coverage_R$, meaning that they score most of the peaks reported by their methods against the reference.

In the first approach, the $F1_p$ score and other measures of classification performance are based on comparing the pairwise alignments and can therefore give only limited information about the full multiple alignment performance due to the exclusion of double and longer gaps in the pairwise alignment comparison across samples. Additionally, their total number for TP, FP, FN and TN are much higher than ours, since they count each peak list multiple times when all pairwise alignments are evaluated. With our row-wise counting method, these double or longer gaps do not hamper the calculation of the performance numbers, since the minimum number of reported peaks in a row is two, which always allows to map reported peak groups against the correct reference group. Thus, we have a one-to-one mapping of peaks as reported by each tool against those contained in the reference alignment and a single performance number in contrast to having to define an average (here termed $F1_p$) with standard deviation over the pairwise alignments for each algorithm's results.

Since both approaches are viable methods to quantify the performance of the alignments, giving weight to slightly different aspects, we have additionally mentioned and referenced the best average results for $F1_p$ in the main manuscript and included the corresponding charts in this section for each dataset. Further tables and plots of precision and recall, as well as FPs, TPs, FNs, and TNs are provided in Supplementary Material 2 of Hoffmann et al. (2014).

Acknowledgements

First and foremost, I would like to thank my supervisors, Prof. Dr. Jens Stoye and Prof. Dr. Karsten Niehaus for their continued support and supervision during the work on this thesis. I have often profited from their constructive criticism and scientific expertise during many fruitful discussions.

I enjoyed sharing my various offices with a number of great people while being a member of the Genome Informatics Group. I would like to thank Wolfgang Gerlach for interesting and insightful discussions on various topics and for his friendship. For the last two years, I shared my office with Pina Krell and Christina Ander, both of whom were excellent officemates and colleagues, sharing ups and downs and a lot of know-how. I would also like to thank all the other members of the Genome Informatics Group, past or present, who crossed my way and made a lasting impression. However, without Heike, the group would not be complete. Thank you for your support, both administrative and personal.

As an associated member of the International Graduate School in Bioinformatics and Genome Research, I would like to express my gratitude for receiving funding in order to attend a number of international conferences. I would especially like to thank the former executive director, Dr. Susanne Schneiker-Bekel and the former secretary, Silke Kölsch, for their administrative help and occasional encouragement. Additionally, I would like to thank all fellow PhD students of the graduate school for their support and feedback, especially after the regular PhD project presentations.

I would like to thank all students that have helped me realize MALTCMS and MAUI during their Bachelor Theses, during projects, or as student research assistants: Mathias Wilhelm, who contributed to MALTCMS and CHROMA4D and early prototypes of MAUI, Rolf Hilker for his help on implementing the metabolite database user interface, Kai-Bernd Stadermann for his work on MPAXS, Sören Müller for his work on early prototypes of MALTCMS and TIC-based alignment, and Konstantin Otte for his implementation help and feedback on core features of MAUI that are

shared with the proteomics software PROTEUS that he developed during his master studies' project.

A number of people provided datasets and biological, computer-science, as well as chemical expertise that helped tremendously in developing, evaluating, and test-driving both MALTCMS and MAUI.

I further wish to thank Dr. Steffen Neumann at the Institute for Plant Biochemistry in Halle/Saale for his continued expert advice, friendship, and fruitful collaboration on metabolomics data standardization.

I would also like to thank Manuela Meyer, formerly at the Center for Biotechnology (CeBiTec), for the preparation of the GC-MS samples used in the example workflow for CHROMA. I furthermore thank Denise Schöfbeck and Dr. Rainer Schumacher, at the Center for Analytical Chemistry, IFA Tulln, Austria, for measuring those samples and for kindly providing the datasets.

I am very grateful for the support of Dr. Petra Högy at the Institute for Plant Ecology and Ecotoxicology, Hohenheim University, who provided the Wheat dataset that was used in the BIPACE evaluation. Additionally, I am grateful for great GC×GC-MS chromatograms and scientific support provided by Dr. Mathias Keck (formerly Metabolomics and Proteomics Group, CeBiTec), who was also involved in acquiring the Wheat dataset samples.

Furthermore, I would like to express my gratitude to Anja Döbbe and Olaf Kruse, at the Department for Algae Biotechnology and Bioenergy at the CeBiTec, for providing the GC×GC-MS samples and datasets used in the BIPACE 2D evaluation, the example CHROMA4D workflow, and the manually annotated reference multiple alignment.

I would like to thank Theresa Strätner for her experimental advice and both her patience and feedback during the development of MAUI. I am also grateful to Henning Kuich at the quantitative proteomics and metabolomics platform group at the Max-Delbrück-Centrum for molecular medicine in Berlin for applying MAUI and MALTCMS outside of Bielefeld and for providing helpful feedback and ideas. I further thank Dr. Dietrich Meier at the Thünen Institute of Wood Chemistry, Hamburg for test-driving MALTCMS AP and for fruitful discussions and valuable feedback.

Also, I would like to thank everyone who helped me with proofreading of this thesis, especially Fábio Martinez, Henner Sudek, Anja Doebbe, and Kai Bernd Stadermann.

There are many other people at the CeBiTec and at the Faculty of Technology who I would like to thank for their help, collaboration, feedback, and coffee, but there are just too many to name all of them individually.

And finally, most important of all, I would like to thank my wife Esther and my sons Arne and Erik and all other members of my family for their loving support and continued patience.

Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbständig angefertigt und keine weiteren als die angegebenen Hilfsmittel und Quellen verwendet zu haben.

Bielefeld, 26. Mai 2014 .....

Nils Hoffmann