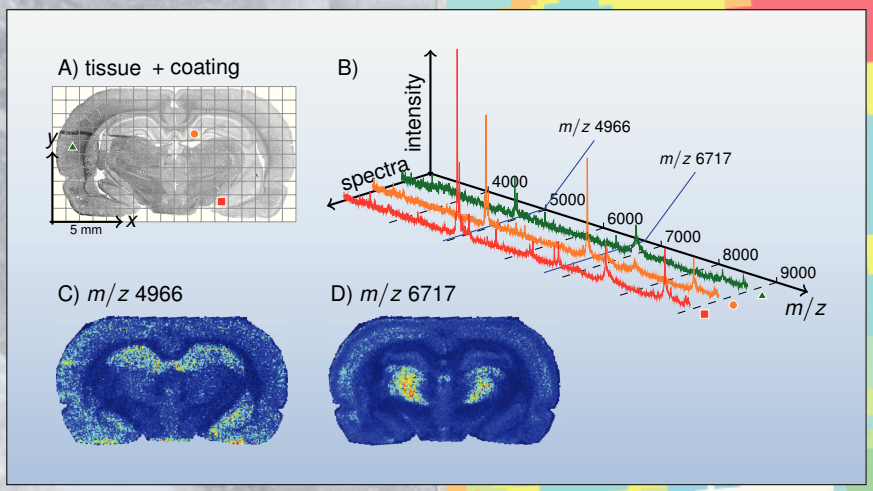
# Signal and image processing methods for imaging mass spectrometry data

Jan Hendrik Kobarg



# Signal and image processing methods for imaging mass spectrometry data

Jan Hendrik Kobarg

**Dissertation**
zur Erlangung des Grades eines Doktors der Naturwissenschaften
– Dr. rer. nat –

Vorgelegt im Fachbereich 3 (Mathematik & Informatik)
der Universität Bremen
im August 2014

Datum des Promotionskolloquiums: 26. September 2014

Gutachter:
Prof. Dr. Peter Maaß (Universität Bremen)
Dr. Theodore Alexandrov (Universität Bremen)

***Knowledge*** *is*
*having the right answer.*

***Intelligence*** *is*
*asking the right question.*

# Contents

# Preface

*Rome was not built in a day* is probably the best way to describe this thesis. Needless to say that it would not be the same without the help and input from my colleagues at ZeTeM and SCiLS. Extra special thanks to Andreas Bartels, Lena Hauberg-Lotte, Oliver Keszöcze, Andrew Palmer, and Klaus Steinhorst for proof reading and bringing order into my written thoughts.

I want to thank Peter Maaß and Theodore Alexandrov for supervising my research and reviewing my thesis. Also to the European Commission's FP7 project UNLocX which indirectly funded my research.

Outside of the university I want to thank everybody was there to provide diversion from the work. Especially my parents Klaus and Sabine for all the great support.

Jan Hendrik Kobarg
Bremen, August 2014

# 1 Introduction

## 1.1 Motivation for this thesis

In recent years *imaging mass spectrometry* (IMS) has evolved as an analysis tool for many biological applications. In contrast to classical mass spectrometry, the data provides not only molecular information in the form of $m/z$ values, but also spatially resolved information, as shown in Figure 1.1. However, converting the data into information requires efficient computational methods (Jones et al., 2012a). Several approaches to analyse the data have been proposed (Bonnel et al., 2011; Hanselmann, 2010; Lee and Gilmore, 2009), among which segmentation of the dataset with clustering algorithms is most prominent (Chaurand et al., 2004; McCombie et al., 2005). In this thesis, the focus lays on the processing pipeline by Alexandrov et al. (2010) which performs spatial smoothing before clustering.

Despite the fact that several groups are working on developing computational methods for IMS, their work is evaluated on real-life data, where the ground truth is not known. When the ground truth is not known, it is difficult to reliably or quantitatively validate new algorithms. So far some approaches utilize statistical simulation to validate their work, but the majority uses real-life data. For analysis ofIMS data produced by *matrix assisted laser desorption/ionization* (MALDI) *time-of-flight* (TOF) mass spectrometer, physical and statistical simulation models have been developed for single spectrum MS (Coombes et al., 2005; House et al., 2011). In this thesis, the existing simulation models are combined and expanded such that one can obtain IMS data of arbitrary size.

Since the technique of IMS is gaining popularity, computational problems arise when the established processing steps are applied to new data. Technical improvements occur, for example the rate at which spectra of two dimensional tissue sections can be obtained. The number of acquired spectra was relative low before usage of IMS and so most computational algorithms only work with small numbers of spectra. However, when it comes to large dataset sizes the naive use of most algorithms fails. The computational effort grows with the number of spectra

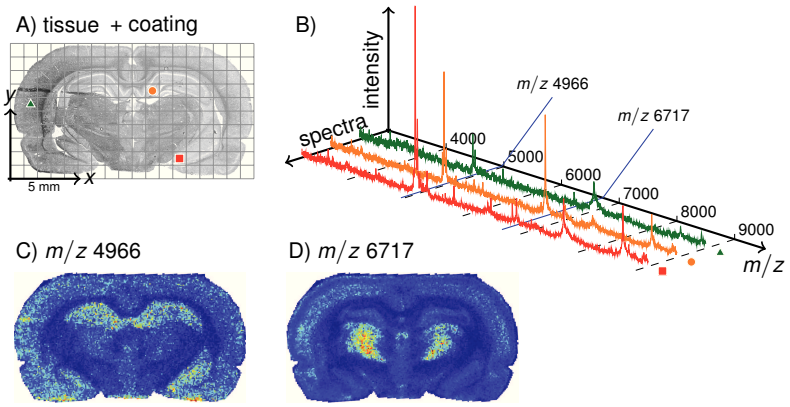



**Figure 1.1: The idea of imaging mass spectrometry.** A. On a tissue section a predefined grid defines different spatial locations. B. Spectra are acquired at the spatial locations. C-D. Once the entire data is loaded, $m/z$ images can be created. (Reproduced from Alexandrov and Kobarg, 2011.)

considered for processing, requiring long run-times and high memory need. Moreover, it is feasible to acquire three dimensional datasets (Crecelius et al., 2005). Where a series of sections from the same tissue can be measured and then digitally reconstructed to the original physical structure. Recently, a protocol was developed by Trede et al. (2012b) and later computationally refined by Oetjen et al. (2013) which proposes to use the bisecting $K$-means for fast top-down segmentation of large datasets.

Several more small modifications like these to the established processing pipeline for IMS data can be thought of. In this thesis, two alternative approaches are considered. First, different types of matrix factorization as a way to reduce the size of the data are considered and compared. Therefore, *principal component analysis* (PCA; McCombie et al., 2005) and *non-negative matrix factorization* (NMF; Jones et al., 2011) are described and NMF is improved to be more suitable for the spatial nature of IMS data (Kobarg et al., 2014). Instead of choosing methods that ease the interpretation of the data, efficient data compression methods that decrease computation time can also be considered. Here the method of FastMap (Faloutsos and Lin, 1995) is used.

Edge-preserving spatial smoothing has a positive effect on the segmentation of IMS data (Alexandrov et al., 2010), but is time-consuming step. Therefore, alternative methods should be considered that are fast. The field of image processing (Bredies and Lorenz, 2011) offers a wide range of tools to select from. Moreover, it will be shown that the ideas of spatial smoothing and efficient data compression can be combined to result in a new data representation. This new data

representation is directly suitable for segmentation by clustering methods while containing the spatial information especially present in IMS data (Alexandrov and Kobarg, 2011).

Due to the increasing number of spectra in an imaging dataset as well as the increase of data points within a single spectra, efficient algorithms have to be considered not only for processing, but also the algorithms have to be examined for their capability to be applied in parallel (Jones et al., 2012b). Parallelizable algorithms can be run on a *graphics processing unit* (GPU). Doing computation on the GPU accelerates computation time in the order of several magnitudes (Chen et al., 2007; Hussong et al., 2009). Naturally, an efficient use of processing time requires the task is fulfilled with GPU-friendly algorithms (Chen et al., 2007; Pock et al., 2008). In this thesis, the candidates suitable for GPU processing are identified for efficient baseline removal, as well as spatial smoothing.

## 1.2 Overview of imaging mass spectrometry

Mass spectrometry is a technique of analytical chemistry used to reveal information of molecular composition (Colinge and Bennett, 2007; Liu et al., 2003; Vitek, 2009). This information is of interest in biology and medicine, which are two important fields of application for mass spectrometry (Aebersold and Mann, 2003; Schwartz et al., 2004). Obtaining information not only about the full chemical composition of the sample, but also about its spatial location gives name to IMS (Caprioli et al., 1997; Stoeckli et al., 2001; Watrous et al., 2011). In an IMS experiment a spectrum is acquired and stored for each pixel of the tissue (see Figure 1.1A). Usually it is impractical to view all the spectra (see Figure 1.1B), as such a dataset consists of several tens of thousand pixels. Instead, once the entire data has been acquired, an $m/z$ image is constructed. In such $m/z$ images the intensity of the given $m/z$ value is visualized by colour. Two representative $m/z$ images are shown in Figure 1.1C-D revealing intensities for two different $m/z$ values in different anatomical regions.

Numerous studies with IMS allow for better understanding of the chemical composition and biological processes, as described in recent reviews (Heeren, 2014; Römpp and Spengler, 2013; Amstalden van Hove et al., 2010; Watrous et al., 2011) and several dissertations (Balluff, 2013; Wehder, 2013; Meding, 2012; Hanselmann, 2010; Broersen, 2009; Lange, 2008; Shin, 2006). IMS has proven its potential in discovery of new drugs (Yang et al., 2009; Solon et al., 2010), cancer biomarkers (Cazares et al., 2009; Rauser et al., 2010b), and protein identification (Jungmann and Heeren, 2012), just to mention a few important applications.

Several different approaches for the acquisition of mass spectra exist. However, just a few can be applied to represent the data as an image. In surface analysis, *secondary ion mass spectrometry* (SIMS, Benninghoven and Loebach, 1971) is most often used. Measuring small and large molecules (e.,g. metabolites, lipids, and proteins) is possible with *matrix assisted laser desorption ionization* (MALDI, Karas and Hillenkamp, 1988, Tanaka et al., 1988). The analysis of mass spectrometry data is normally organized in several phases: The first phase is the acquisition of spectra to form the dataset. The data obtained in IMS experiments is large. Datasets are comprising $10^4$ pixels in the 2D case and easily reach more than $10^6$ pixels in 3D experiments. For each pixel, the measured spectrum usually contains $10^3$ to $10^5$ data points. With this massive multivariate dataset, it is challenging to find the important information. Therefore, the acquisition phase is followed by computational processing. Computational exploration of multivariate data suffers from the curse of dimensionality (Hastie et al., 2009) and several approaches have been proposed to counter this effect. Popular choices are dimensionality reduction with PCA and then either hierarchical clustering (Deininger et al., 2008) or clustering with $K$-means (McCombie et al., 2005) of features obtained by PCA. An efficient processing pipeline was introduced by Alexandrov et al. (2010) which includes spatial smoothing to remove the high pixel-to-pixel variation of intensities. Based on this processing pipeline, several modifications are introduced (Alexandrov and Kobarg, 2011; Trede et al., 2012b) and described in the later chapters of this thesis. Recently, also matrix factorization methods with NMF were considered (Kobarg et al., 2014; Jones et al., 2012a).

### 1.2.1 Acquisition of data using matrix assisted laser desorption/ionization

Before the acquisition of spectra for IMS can take place, a sample preparation involves cutting the tissue and mounting it on a conductive coated glass slide (Römpp and Spengler, 2013). The section thickness is usually in the order of 10-15 µm for frozen tissue and few µm for formalin fixed tissue (Lemaire et al., 2007; Gustafsson et al., 2010; Casadonte and Caprioli, 2011; Trede et al., 2012b). Different embedding materials can be used with IMS such as carboxymethyl cellulose (Kawamoto, 2003), gelatine (Altelaar et al., 2005), a polymer compound (Strohalm et al., 2011), and tragacanth gum (Brignole-Baudouin et al., 2012). For control of acquisition, and visualization, an optical image of the tissue section is obtained, see Figure 1.2 (left). In the next step, sample preparation for the detection of proteins uses a bath of organic solvents to remove molecules which otherwise suppress protein ionisation. This step is called washing and will later improve the spectrum quality as extra molecules such as lipids are removed from

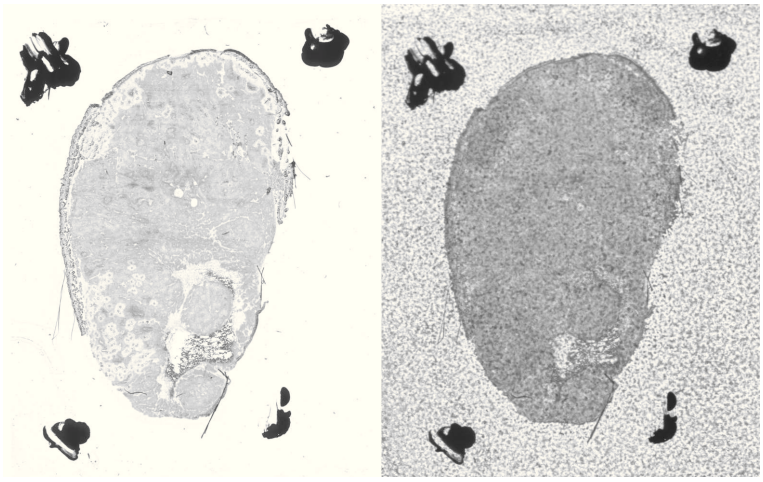


**Figure 1.2: Optical image of tissue section prepared for MALDI imaging.** Reference image used to control data acquisition (left); image of same section after application of $\alpha$-cyano-4-hydroxycinnamic acid (right); images provided by MALDI Imaging Lab, University of Bremen.

the tissue (Schwartz et al., 2003). For a MALDI experiment, which is focused on optimal spatial resolution, a *matrix solution* is deposited onto the tissue. This can be done in multiple ways, like with acustic deposition (Aerni et al., 2006), inkjet printers (Baluya et al., 2007), sublimation (Hankin et al., 2007), or spraying (Schwartz et al., 2003). Typically the matrix is a weak organic acid that when brought onto the tissue will crystallize and form a layer on the tissue. Such a layer promotes the soft ionization of intact molecules from the tissue (Karas and Hillenkamp, 1988). The optical effect of the dried crystallized layer is shown in Figure 1.2 (right). Sometimes the research interest is additional solvent extraction at the expense of spatial resolution. In this case larger droplets of matrix are applied to the tissue (Schwartz et al., 2003). In this thesis the first case will mainly be considered. Common matrix solutions contain 3,5-dimethoxy-4-hydroxycinnamic acid (commonly known as sinapinic acid, SA), $\alpha$-cyano-4-hydroxycinnamic acid (CHCA), or 2,5-dihydroxy benzoic acid (DHB), which are commercially available (Schwartz et al., 2003). They are applied to the tissue depending on the interest of research. In comparison to the molecules one intends to probe the matrix consists of much lighter molecules. The light molecules surround the heavy molecules once the matrix has crystallized on the tissue, see Figure 1.3 for a schematic display.

IMS samples that are prepared in such a manner are ready to be analysed in a mass spectrometer. A mass spectrometer consists of two essential parts: an ion source and a mass analyser with

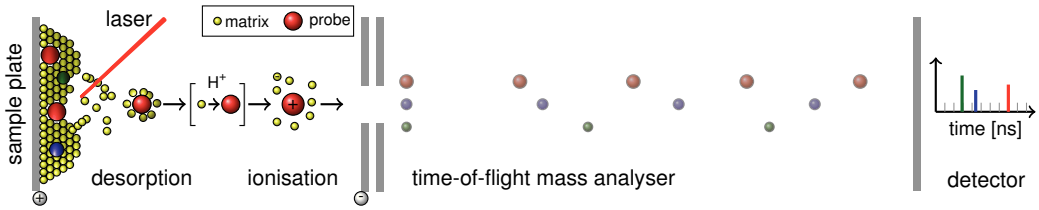



**Figure 1.3: Matrix assisted laser desorption/ionization with time-of-flight detection.** Molecules are enclosed by light matrix molecules of an acid, the matrix absorbs the laser energy and removes the heavy molecules from the tissue. An electric field accelerates the charged molecules into the direction of the time-of-flight mass analyser. The detector at the end of the mass analyser converts the count of arriving ions into a line spectrum.

a detector. In the following the ion source is considered to be MALDI (Karas and Hillenkamp, 1988; Tanaka et al., 1988) and the analyser TOF. In this setting, a laser fires at the matrix crystals (see Figure 1.3). The energy of the laser is absorbed by the matrix and a plume of material is desorbed from the tissue. As the matrix absorbs most of the energy, the heavy molecules stay intact and are usually not fragmented into several parts (Knochenmuss and Zhigilei, 2005; Karas and Hillenkamp, 1988; Tanaka et al., 1988). The ionized material is accelerated into a flight tube into the direction of the detector. In the case of the TOF analyser the different molecules all have the same energy, but due to their different masses have different accelerations, this results in different velocities. After travelling through the flight tube they reach the detector mounted at the end of the tube in different time. The detector records the number of ions reaching it within a defined interval. The interval is called the *detector bin* in the context of mass spectrometry. Knowing the flight time for a set of calibration molecules, the mass-to-charge ratio $m/z$ can be computed. In MALDI molecules have usually single charge that is why the ratio expresses the mass.

The overall goal of computational analysis is to determine important $m/z$ values from the thousands of peaks generated automatically. With candidate $m/z$ values found, follow-up experiments can be performed that aim to identify the molecules. For this, specialized MS/MS experiments need to be run, see for example Aebersold and Mann (2003) for a description of protein or Gustafsson et al. (2012) for peptide identification.

As already stated in this thesis, the IMS data considered was acquired with linear TOF analyser. Usually, linear TOF for imaging provides short spectra with $10^3$ to $10^4$ data points per spectrum (McDonnell et al., 2010). As well as linear TOF, reflector instruments exist, where the

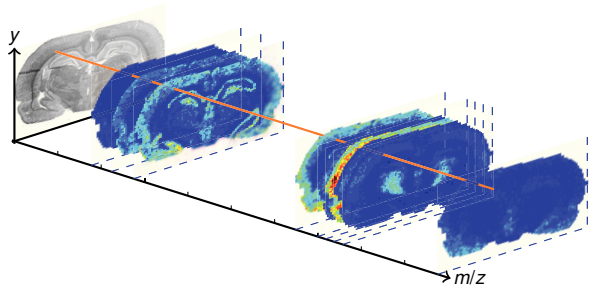



**Figure 1.4: Hyperspectral image cube interpretation.** The stack of $m/z$ images reduced to an important list of images that have been identified by peak picking.

flight path is doubled. This doubled flight path increases the mass resolution and allows to resolve isotopic peaks, but requires more detector bins to be stored (Goodwin et al., 2008). Other types of mass analyser also exist such as the *Fourier transform ion cyclotron resonance* (FTICR). FTICR mass analyser have a higher mass resolution in the order of several magnitudes with the number of data points exceeding $10^6$ (Cornett et al., 2008). Only few algorithms presented in this thesis have been applied to high resolution data. However, no theoretical restriction on the type of data was made at any point. Furthermore, most algorithms work on a reduced set of important peaks rather than the full spectrum. With such peak lists it does not make a difference if they result from high resolution spectra or not. Moreover, other research areas that use hyperspectral data can easily adopt the algorithms for their own application.

### 1.2.2 Computational aspects for imaging mass spectrometry data

In this thesis, the processing steps for IMS proposed by Alexandrov et al. (2010) will be the main theme. Later chapters will describe improvements and explore them in more detail. All of the proposed steps involve the elementary phases of preprocessing and clustering of the data into similar groups. In most cases, the obtained spectra must first be *normalized* to a similar intensity range. Normalization is necessary due to the fact that in different areas of the tissue the number of molecules removed from the tissue will differ (Deininger et al., 2011). Furthermore, the spectra will contain an intensity offset that is called *baseline* (Williams et al., 2005). The baseline is related to the applied matrix (Schwartz et al., 2003). Usually it is similar within a sample, but it is also known to be dependent on the spatial location (Norris et al., 2007).

It is important to remove redundant information from the data. In the context of mass spectrometry, this step is often called *peak picking* or *feature extraction* (Foley, 1987; Coombes et al.,

2003). Generally speaking, peak picking is a way of dimensionality reduction. The redundancy originates mainly from the length of the spectra. Feature extraction has one important advantage: the processing time for feature extraction usually is much shorter than the time needed for further processing the complete data. A grouping of the spectra in the dataset is then achieved by spatial segmentation with clustering algorithms (Chaurand et al., 2004; McCombie et al., 2005). Another approach that is recently used (Kobarg et al., 2014; Jones et al., 2012a) makes use of matrix factorization methods which will also be covered in this thesis.

A complete dataset that is reduced to important peaks can be thought as a stack of $m/z$ images as shown in Figure 1.4. Due to several effects of IMS the quality of the spectra differs from pixel to pixel. Therefore, image processing methods for denoising are applied to the dataset and exploit its spatial nature. The range of denoising methods is huge as for different approaches specialized methods were created (Bredies and Lorenz, 2011). Especially good results are obtained by using edge preserving denoising techniques applied to each of the $m/z$ images in the data (Alexandrov et al., 2010). Trede et al. (2012b) use the iterative edge preserving denoising algorithm by Chambolle (2004) while this thesis considers a fast bilateral filter by Tomasi and Manduchi (1998) which also preserves edges. As an alternative to the channel-by-channel smoothing techniques, embedding the data into a spatially aware feature space was proposed by Alexandrov and Kobarg (2011). The feature space is of a higher dimension such that the segmentation by linear hyperplances becomes highly effective.

### 1.2.3 Description of datasets used in this thesis

The following is a summary of the datasets used in this thesis. Data acquisition has been performed with standard measurement protocols. Sample preparation depends on the type of data and will be described in the respective section.

#### Coronal rat brain dataset

This dataset has been used by Alexandrov et al. (2010), Alexandrov and Kobarg (2011) and Alexandrov and Bartels (2013) with different research aims. Rodent organs are standard examples in IMS research (Watrous et al., 2011), because they are well annotated and easy to interpret even for non-biologists. A schematic of the anatomical structures in this dataset is shown in Figure 1.5. The original data was provided by Michael Becker (Bruker Daltonik GmbH, Bremen, Germany) and first used by Alexandrov et al. (2010) where spatial smoothing was introduced

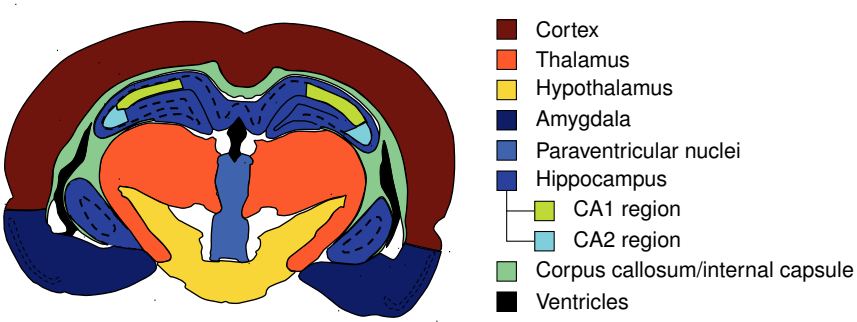



**Figure 1.5: Schematical display of the coronal rat brain dataset.** The different anatomical regions with annotations. (Reproduced from Alexandrov and Kobarg, 2011.)

to improve spatial segmentation. Spatial segmentation maps were then employed to discover relevant $m/z$ values in the data. Furthermore, Alexandrov and Kobarg (2011) used the dataset to demonstrate an improved peak picking method and to compare the effectiveness of different segmentation algorithms. By Alexandrov and Bartels (2013) manual grouping of $m/z$ values into a test set was performed.

Sample preparation was performed in the following way: The rat brain was cut on a cryostat into sections of 10 µm thickness and transferred to a precooled, conductive indium-tin-oxide (ITO) coated glass slide (Bruker Daltonik GmbH, Bremen, Germany). The section was washed twice for 1 min in 70% ethanol, and once for 1 min in 96% ethanol and then dried in a vacuum desiccator. The matrix (Sinapinic acid; 10 mg/mL in 60% acetonitrile and 40% water with 0.2% trifluoroacetic acid) was applied using the ImagePrep device (Bruker Daltonik GmbH) following a standard protocol. Mass spectra were acquired on a MALDI-TOF instrument (Autoflex III; Bruker Daltonik GmbH) equipped with a 200 Hz smartbeam II laser. MALDI measurements were performed in linear positive mode at a mass range of 2.5 to 25 kDa. The lateral resolution for the MALDI image was set to 80 µm. A total of 200 laser shots were summed up per position. The rat brain dataset comprises 20,185 spectra acquired within the section area ($120 \times 201$ pixels), each of 6618 data points.

### Neuroendocrine tumor dataset

Another dataset used in this thesis is from a section of a neuroendocrine tumor (NET) invading the small intestine (ileum). It was used by Alexandrov et al. (2010) and Alexandrov and Kobarg (2011) to compare algorithm performance for a highly complex tumor tissue sample. A

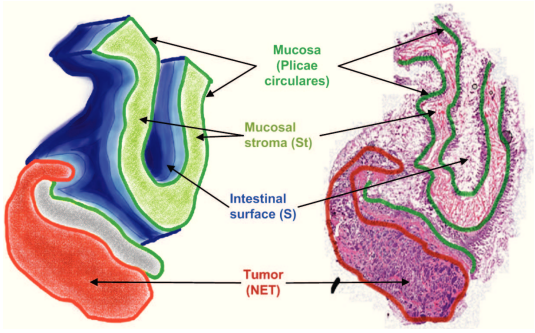



**Figure 1.6: Schematic of neuroendocrine tumor dataset.** Important histopathological regions present in the section. (Reproduced from Alexandrov et al., 2010.)

schematic of the anatomical structures in this dataset is shown in Figure 1.6. Here the sample preparation is reprinted from Alexandrov and Kobarg (2011): The sections were washed twice for 30 s in 70% ethanol, and once for 20 s in 96% ethanol, and then dried in a vacuum desiccator. The matrix was applied in the same way as for the rat brain sample. Mass spectra were acquired on a MALDI-TOF instrument (Autoflex III, Bruker Daltonik GmbH) equipped with a 200 Hz smartbeam II laser. MALDI measurements were performed in linear positive mode at a mass range of 1 to 30 kDa with a lateral resolution of 50 µm and 300 laser shots per position. For data processing, only the mass range from 3.2 to 18 kDa was considered. After MALDI analysis, the matrix was washed off using 70% ethanol, and a conventional hematoxylin and eosin (H&E) staining was performed. The stained sections, coregistered with the MALDI-imaging results, were evaluated histologically by an experienced pathologist using a virtual slide scanner (MIRAX desk, Carl Zeiss MicroImaging GmbH, Munich, Germany). The dataset comprises 27,360 spectra each of 5027 data points.

### Spinach single spectrum dataset

In order to explore the effect of spectral resolution, sample spectra of a piece of crude spinach were measured (Kobarg et al., 2014). Pigments were isolated from fresh frozen spinach leafs with repeated cycles of a methanol extraction and then filtered. 0.5 µl of the extract was spotted on a stainless steal MALDI sample stage and mixed with 0.5 µl of the respective matrix solution according to the dried-droplet method (Karas and Hillenkamp, 1988). Only a single spectrum per available repetition rate (4 GHz, 2 GHz, ..., 62.5 MHz) was obtained in both positive linear

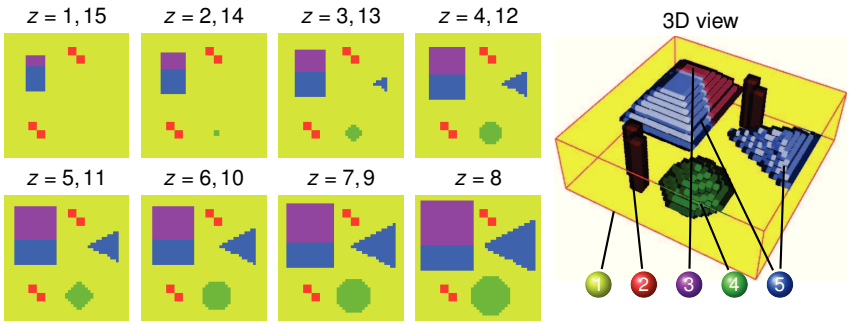



**Figure 1.7: True class image of simulated dataset**. Virtual slices for a fixed spatial value $z = 1,\ldots,15$, and the corresponding 3D view. Dimension of the dataset is $45 \times 45 \times 15 = 30{,}375$ voxels, 5 classes, $m = 6639$ channels. (Reproduced from Kobarg and Alexandrov, 2013.)

mode and positive reflector mode, each for the mass range of 800 to 1000 Da. The setup of this experiment allows to detect the isotope peaks by identification in the high sampled data.

### Benchmark 3D datacube

Kobarg and Alexandrov (2013) simulated a dataset with $n = n_x \times n_y \times n_z = 45 \times 45 \times 15$ spectra located at three spatial dimensions $(x,y,z)$ represented as *voxels*. The spectra belong to $K = 5$ classes which form simple geometric shapes, see Figure 1.7. These objects are background (1), sticks (2), upper rectangle (3), ball (4), pyramid (5a), and lower rectangle (5b), with the last two belonging to the same class. Based on the true class assigned to the voxel, each spectrum was simulated independently of other voxels. For each class a template spectrum with $m = 6972$ channels was selected from a real-life dataset of rat kidney. The selection of the template spectra was based on an initial segmentation of the unsmoothed dataset into five classes. Those five spectra that were closest to the computed class means were used as templates. The abundances at $p = 145$ peak positions found in the template spectra were used within this class. As in the real world, the peak positions vary by small differences in atom mass which add up to mass offset errors. This effect is simulated by using the physical model of particles traversing a flight tube (Coombes et al., 2005). MALDI-TOF spectra have the peaks superimposed to a noise level that forms a baseline. Such a baseline can be approximated by the sum of two exponential functions which in turn are characterized by four parameters in total. For the baselines found in the five template spectra estimation of their parameters is done according to the method described by House et al. (2011). These parameters are taken as estimates of the true parameters for baselines

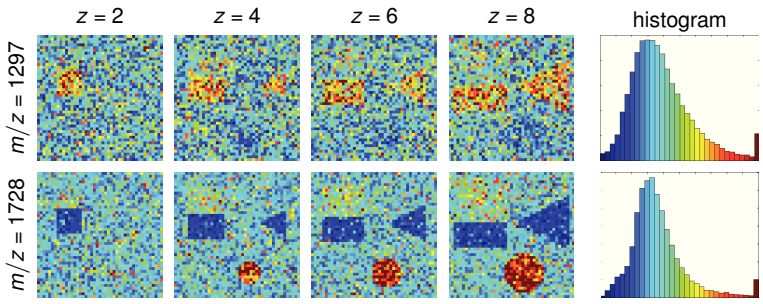



**Figure 1.8: Two selected channel images, along with histogram of intensities.** (Reproduced from Kobarg and Alexandrov, 2013.)

in the dataset. In the last step, the function is used as the estimate of the general noise level in the $m/z$ channels of the spectrum. The effect of noise can be seen in Figure 1.8, where two channels images are displayed.

### Three-dimensional mouse kidney dataset

A single section from an organ gives only a restricted view on spatial distributions within the whole organ. Three dimensional experiments where for example carried out by Crecelius et al. (2005) and Andersson et al. (2008). In this thesis the dataset acquired by Trede et al. (2012b) is used. The spectra acquired in individually measured tissue sections need to be assigned to coordinates within the three dimensional structure of the organ. Access to the coordinates can be gained by using image registration methods applied to the optical images of the tissue sections. Reconstruction with state-of-the-art registration methods requires a set of high quality optical images and few damages to the tissue during cutting. Damaging the tissue can be reduced by embedding the organ (Goodwin, 2012). Having the original coordinates within the three dimensional organ allows co-registration of the spectral information with other imaging modalities, such as *magnetic resonance imaging* (MRI; Glunde et al., 2009) For a detailed description of the registration methods and visualization with imaging modalities, see Thiele et al. (2014).

The acquisition protocol for this particular dataset is described in detail by Trede et al. (2012b). In short, the whole tissue had to be fixed in paraffin using the PAXgene tissue fixation reagent (PreNalaytiX GmbH, Germany) to not damage the three-dimensional shape during cutting. The central part of the mouse kidney was sliced into sections of 3.5 µm thickness. A representative set of 33 sections was selected and for each serial section, a 2D MALDI-IMS dataset was acquired. Acquisition followed the standard protocols of matrix application and spectrum ac-

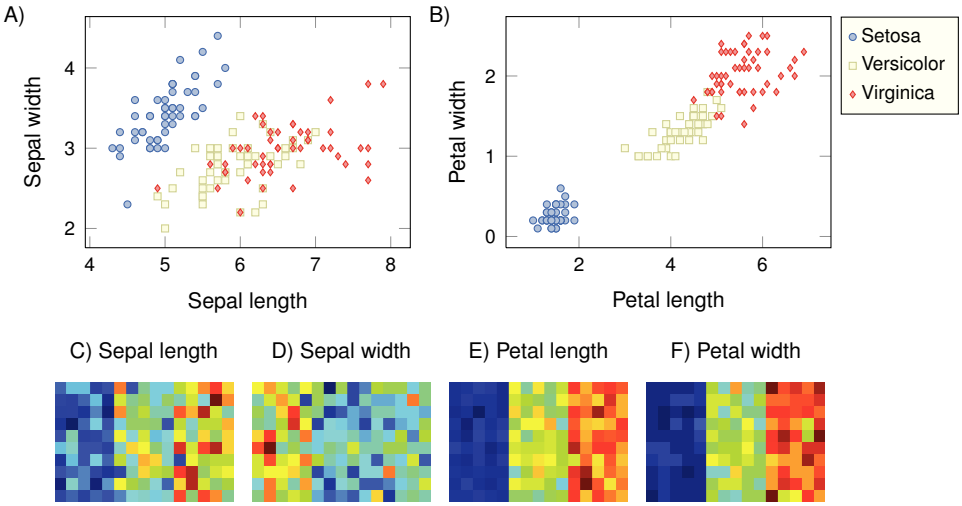



**Figure 1.9: Plot of Fisher's iris multivariate dataset.** A-B. Covariables plotted against each other in a scatter plot. C-F. visualized as channel images.

quisition as described in Section 1.2.3. Additionally, the lateral resolution was increased to 50 µm and 300 laser shots per position were performed. In total there are 512,495 spectra for the 33 sections, where each spectrum comprises 7677 $m/z$ values such that the size is 50 GB.

A second mouse kidney used by Oetjen et al. (2013) contains four times the data: in total there are 122 sections with 2,171,451 spectra. Here each spectrum comprises 7680 $m/z$ values making the dataset 200 GB.

### Fisher's iris dataset

As a reference dataset to show some simple concepts in multivariate statistics, the Iris dataset collected by Anderson (1936) is used here as well. It is most popular due to the publication by Fisher (1936). The data originates from three different species of iris flowers, *Iris setosa*, *Iris versicolor*, and *Iris virginica*. For each of the species a sample of 50 flowers was examined by measuring the length and the width of the sepals and petals. The dataset is of special interest since it constitutes a difficult separation problem. While the species *I. setosa* can be separated based on the sepal dimensions, the lengths of the *I. versicolor* and *I. virginica* species are too similar for linear separation. See Figure 1.9 for the paired plot and spatial representation as channel images. Originally, there is no information how the plants were cultivated, except they

are from the same colony. However, when one assumes the flowers where aligned in a field where they were grouped by their species, one can create an imaging dataset. This way, the intensities in each channel and clustering results are easy to visualize.

## 1.3 Previous contributions and scope of this thesis

Large portions of this thesis build up onto the work by Alexandrov et al. (2010). Parts of the research presented in this thesis can also be found in the following publications:

- Alexandrov, T. and Kobarg, J. H. (2011): "Efficient spatial segmentation of large imaging mass spectrometry datasets with spatially aware clustering". *Bioinformatics*, 27(13): i230–i238.
- Kobarg, J. H. and Alexandrov, T. (2013): "Efficient spatial segmentation of hyperspectral 3D volume data". In: *Algorithms from and for Nature and Life*. Ed. by B. Lausen, D. van den Pol, and A. Ultsch. Studies in Classification, Data Analysis, and Knowledge Organization. Switzerland: Springer International Publishing, pp. 95–103.
- Kobarg, J. H., Maass, P., Oetjen, J., Tropp, O., Hirsch, E., Sagiv, C., et al. (2014): "Numerical experiments with MALDI Imaging data". *Advances in Computational Mathematics*, 40(3): 667–682.

Furthermore, this thesis is based on a series of work package reports from the European Commission funded FP7 project "Uncertainty principles versus localization properties, function systems for efficient coding schemes" (UNLocX):

- Maass, P., Kobarg, J. H., and Thiele, H. (2011a): *A Phase Space Concept for MALDI Data*. Internal Report 7.1. Project UNLocX: University of Bremen.
- Maass, P., Kobarg, J. H., and Thiele, H. (2011b): *Software Specification for MALDI Data*. Internal Report 7.2. Project UNLocX: University of Bremen.
- Kobarg, J. H., Dyatlov, A., and Schiffler, S. (2011): *MALDI Data preprocessing*. Internal Report 7.3. Project UNLocX: University of Bremen.
- Kobarg, J. H., Maass, P., Golbabaee, M., Vandergheynst, P., Tropp, O., and Sagiv, C. (2012a): *Efficient transformation of MALDI data*. Internal Report 7.4. Project UNLocX: University of Bremen.
- Kobarg, J. H. and Maass, P. (2012): *Feature characterization in phase space*. Internal Report 7.5. Project UNLocX: University of Bremen.

- Kobarg, J. H., Tropp, O., Sagiv, C., Rubin, E., and Hirsh, E. (2012b): *GPU implementation of BaseLine Algorithms: Efficient transformation of the MALDI data*. Internal Report 5.4. Project UNLocX: University of Bremen and SagivTech Ltd..
- Kobarg, J. H. and Maass, P. (2013): *Classification with phase space features*. Internal Report 7.6. Project UNLocX: University of Bremen.

The thesis is organized into three major parts. The first part includes the introduction and a brief overview of the computational methods. In Chapter 2 the different aspects of computational processing of IMS data are reviewed to familarize the reader with the topic and establish a common notation for this thesis. For this, a basic mathematical model is defined, that is used within this thesis. The review includes a brief overview of different baseline removal algorithms as well as a method of obtaining peaks from the data as a mean of data reduction. Furthermore, clustering of multivariate data is explained, which is used frequently in the context of IMS. For these clusters $m/z$ values not identified by the peak picking can be retrieved from the data with the method of correlation.

The Chapter 3 contains the second major part. First, individual MALDI mass spectra are modelled for which a review of peak shape functions is performed. Second, a framework for the simulation of IMS datasets is presented. With this framework multiple additional aspects of IMS are introduced which were not considered in existing MS simulators.

The third major part starts with Chapter 4, which focuses on different mathematical methods that are used to further reduce the dimensionality of IMS data. For a definition of terms, the established methods PCA and NMF are reviewed. Then improvements adapted to the special nature of IMS are presented. Furthermore, FastMap as an alternative to both dimensionality reduction methods is presented. FastMap has multiple applications and is useful for efficient implementation of the new spatial segmentation method in the following chapter.

In Chapter 5 multiple new approaches are compared with the established processing steps introduced in Chapter 2. First, a comparison of edge preserving channel-by-channel noise reduction methods of Chambolle and bilateral filter are performed. Second, spatially aware segmentation is presented as a fast method to incorporate the spatial information directly into the data. It was published in Alexandrov and Kobarg (2011) and extended in Kobarg and Alexandrov (2013). This requires the use of the bilateral filter weights combined with the efficient dimensionality reduction from the previous chapter.

The thesis concludes in Chapter 6 where also topics are adressed that could not be covered in this thesis.

# 2 Computational processing of imaging mass spectrometry data

## 2.1 Motivation

This chapter will summarize the methods currently used in the computational analysis of imaging mass spectrometry (IMS) data. In this thesis the individual steps of computational analysis are oriented on the earlier work by Alexandrov et al. (2010). The sequence of analysis steps is called *standard processing pipeline* and is shown in Figure 2.1. The steps of the processing pipeline from (Alexandrov et al., 2010) will be reviewed and establish the common notation for this thesis. It has been used in several publications (Crecelius et al., 2012; Alexandrov et al., 2013; Ernst et al., 2014) and is the overarching topic of this thesis.

Over the years, several preprocessing methods have been developed for mass spectrometry data (Caprioli et al., 1997; Schwartz et al., 2003; Wagner et al., 2003; Prados et al., 2006). The essential part of spectra preprocessing and peak picking is also used when the data is not IMS. Peak picking reduces the data to a meaningful list of $m/z$ values that are relevant to

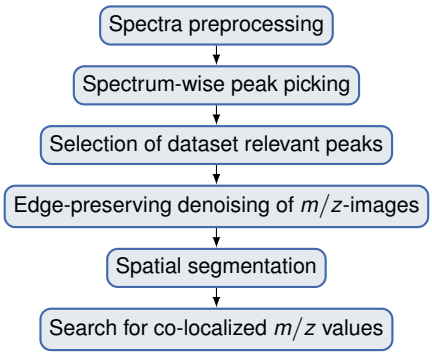



**Figure 2.1: Chart of standard processing pipeline.** The individual steps of the segmentation pipeline as proposed by Alexandrov et al. (2010).

the dataset. Methods specific to the nature of IMS were added later like spatial segmentation (Chaurand et al., 2004; McCombie et al., 2005). Edge-preserving denosing is the youngest addition (Alexandrov et al., 2010) and an important aspect of improving data quality.

Additionally to the concept by Alexandrov et al. (2010), a basic mathematical model for spectra components is the starting point for this chapter. This part is essential to the understanding of mass spectrometry and provides background information needed in Section 3.2. Then a mathematical model for baselines will be described and two algorithms for the removal of the baseline will be introduced. Another concept that is explained in this chapter is the principle of peak picking. The treatment of data with these methods is called preprocessing of the data. Preprocessing is a necessary step to assure data quality for all further processing steps.

The second half of this chapter will introduce common clustering algorithms with agglomerative linkage and $K$-means (Kaufman and Rousseeuw, 1990; MacQueen, 1967). Clustering can be used in IMS to obtain an automatic segmentation of the data. Naturally, the automation requires evaluation methods. Evaluation allows one to access the cluster quality and to compare different segmentations (Rand, 1971; Sebastiani, 2002). Finally, it will be shown how the automatically generated segmentation maps can be used to find interesting masses.

## 2.2 Mathematical model for mass spectra

The obtained dataset can can be represented mathematically in multiple terms. The first representation is along the spatial domain where $X \in \mathbb{R}^{n_1 \times n_2 \times m}$ is a hyperspectral image cube consisting of $n_1 \times n_2$ pixels and $m$ image channels. The image channels relate to the individual detector bins with which the mass spectrum was acquired. In this application the detector bins represent a certain mass-to-charge ratio. Biological samples have arbitrarily shaped border other than a $n_1 \times n_2$ sized rectangle. Accounting only for those $n \leq n_1 n_2$ spectra really measured, the data is simply a collection of intensity vectors $x_i \in \mathbb{R}^m$ with $i = 1, ..., n$ that can be stored in the data matrix

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n \times m} \tag{2.1}$$

or as a set $\mathcal{X} = \{x_i \mid i = 1, ..., n\}$. In both cases the data is represented without considering the spatial location of each object. Although, the spatial information is one of the main aspects in an IMS experiment, the computations are performed in an object-wise manner and the result is

presented in its original spatial location for interpretation. Furthermore, the statistical methods used in this thesis often work only with the non-spatial interpretation of the data. In statistics, all observations are treated as independent from each other and being identically realized from a random distribution (Hastie et al., 2009).

In this thesis and elsewhere (Renard et al., 2008; Morris et al., 2005) the general model of a single mass spectrum

$$s(t) = \sum_{l=1}^{p} a_l f_l(t) + \beta(t) \tag{2.2}$$

as a function of the detector tick $t$ is usually composed of $p$ possibly overlapping *peaks* $f_l(t)$ with *intensity* $a_l$, $l = 1, \ldots, p$, added to a background function $\beta(t)$ called *baseline* . For a spectrum obtained with a perfect mass spectrometer the function $f_l(t)$ would be like the Dirac indicator function $\delta(\mu_l - t)$ for a molecular mass $\mu_l$ and the baseline would vanish to $\beta \equiv 0$. However, it is not possible to obtain the spectra at this perfect quality. Even with high standard equipment the spectrum will be affected by noise. In reality, a convenient assumption is to use a peak shape like a Gaussian distribution

$$f_l(t) = f(t; \mu_l, \sigma_l) = \frac{1}{\sqrt{2\pi}\sigma_l} \exp\left( \frac{(t-\mu_l)^2}{2\sigma_l^2} \right) \tag{2.3}$$

to model this imperfection, where $\mu_l$ and $\sigma_l$ are parameters modelling the mass of the molecule and the standard deviation (Coombes et al., 2005).

As a first approximation this function already works well and in general one assumes fixed standard deviation $\sigma_l = \sigma$ for all $l = 1, \ldots, p$. However, close examination shows that the peak shape function (2.3) is not sufficient (Kempka et al., 2004). For example, the review by Di Marco and Bombi (2001) accounts for 90 different functions to model peak shapes. In general, most methods assume the shape to be essentially a Gaussian, but modify at least its skewness to explain most observable effects. However, the models to describe a skewed Gaussian differ significantly. More importantly, those models only describe peak shapes found in related spectrometry methods, such as surface-enhanced laser desorption/ionization (SELDI) and liquid chromatography (LC-MS). As a consequence, there is no guarantee that the models described in literature will fit to the spectra observed in MALDI mass spectrometry. Different functions for the peak shapes will be the focus of Chapter 3.2.

In an IMS dataset the matrix and molecules are not necessarily equally concentrated in the tissue. Furthermore, all spectra are measured individually and intensities from a mass spectrum

at most account for relative abundances within this spectrum. The normalization step is necessary to improve the spectrum comparability. In comparison to the other preprocessing steps, normalization has a major influence on the interpretability of the data. Deininger et al. (2011) give an excellent review regarding this topic. There are different types of normalization. The most popular choice is the so called *total ion count* (TIC) normalization. Here, one assures the same number of ions in each spectrum, by dividing each spectrum $x \in \mathbb{R}^m$ by the sum of all intensities $\sum_{j=1}^{m} x_j$. This assures that each of the spectra has an ion count of one.

In practice, the usability of any normalization requires expert knowledge of the data (Deininger et al., 2011). If for example some spectra are dominated by high intensities of peaks, while others are not, unwanted and wrong scaling occurs. For more information see Alexandrov (2012) or Sun and Markey (2011).

## 2.3 Mathematical functions for baselines

One can usually observe in MALDI mass spectra, a quite strong baseline effect (Schwartz et al., 2003). This is particularly a problem with direct tissue analysis in IMS in comparison to dried droplets (Norris et al., 2007). In the controlled vocabulary by the HUPO Proteomics Standards Initiative, the term "baseline" is defined as "An attribute of resolution when recording the detector response in absence of the analyte." Several physical explanations for the baseline are given, but are difficult to model and are also influenced by the sample preparations (Sun and Markey, 2011; Schwartz et al., 2003). As theorized by Andrade and Manolakos (2003) as well as House et al. (2011), the measured intensity of the baseline $\beta(t)$ is the major component affected by noise in the form of intensity variations (Kwon et al., 2008). Furthermore, the variance of noise is decreasing with increasing mass in individual spectra (Shin, 2006; House et al., 2011). During the preprocessing of MALDI data, the baseline has to be removed from each spectrum in order to correctly access the peak intensities.

Different approaches for the extraction of the baseline have been proposed. Since the effect of baseline is not well understood, most algorithms propose a heuristic approach. Here, polynomial fitting by Williams et al. (2005) and nonlinear iterative peak-clipping by Ryan et al. (1988) should be mentioned. Points to fit the data to can be obtained by using a robust estimate with the minimum, or a specified quantile, such as the median (Williams et al., 2005). In practice, implementations favoured by instrument manufacturers or major software suites focus on fast algorithms, such as Top-Hat morphological operation (Sauve and Speed, 2004) or MATLAB's

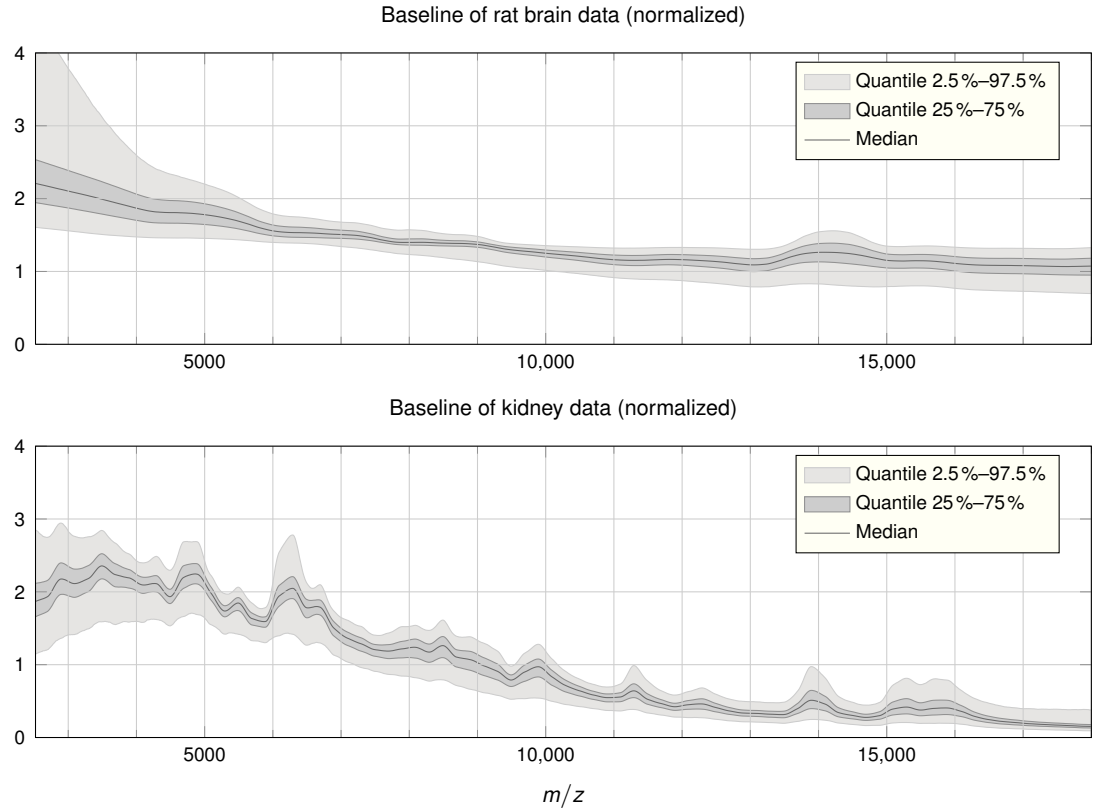



**Figure 2.2: Quantile plot of baselines found in real-life data.** Baselines were obtained with MATLAB for TIC normalized data. Dark grey denotes the quartile range (25%–75%) and light grey denotes the two-sided 5% quantile.

*msbackadj* (Andrade and Manolakos, 2003). The Top-Hat filter is a morphological filter from signal processing (Bredies and Lorenz, 2011). The advantage is that this is a model free way of removing a background signal and uses one parameter to control the window width. With MATLAB's approach, in each window the lower quantile of the signal is found. These values are then used as the estimated contribution of the baseline to the signal. Since the value is only available for each of the windows, interpolation is done through these window estimates.

Andrade and Manolakos (2003) replace the quantile estimate by assuming a probabilistic mixture model to find the mean height of the baseline in each window. Recently, Shin et al. (2010) proposed a different approach to remove the baseline. Here a wavelet denoising is paired with the fitting of an exponential function. Rather than directly accessing the contribution of the

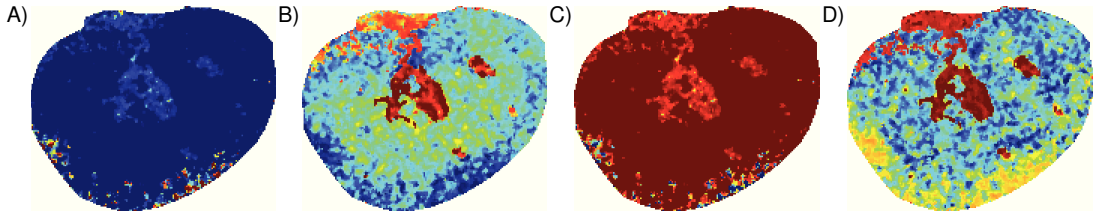



**Figure 2.3: Parameters of the baseline function as spatial image.** The four parameters of the baseline fit plotted for the kidney dataset.

baseline in the signal itself, the interpolation is done in the coefficient space of a decomposed signal. Since wavelets are perfect candidates for parallelization, this approach is of special interest. These two approaches will be examined in further detail.

The plots in Figure 2.2 show quantile information of baselines for the rat brain dataset and a single section from the kidney data computed separately for each bin of the mass spectrum. The baselines were obtained using the *msbackadj* algorithm, which is the baseline removal function from MATLAB's bioinformatics toolbox (Andrade and Manolakos, 2003). The baseline is assumed to account for 10% of the full signal intensity. As can be seen in the quantile plots, the range of baseline becomes narrow with high $m/z$ value. Effectively, this illustrates the observation made by House et al. (2011) and Shin (2006).

A functional approximation for the general trend must be able to fit to several different shapes of baselines. The general trend of the baseline

$$\beta(t) = a_1 \exp(\lambda_1 t) + a_2 \exp(\lambda_2 t) \tag{2.4}$$

can be approximated by the sum of two exponential functions as proposed by Williams et al. (2005) and Shin et al. (2010). This function of four parameters allows to model most effects observed in the data such as the slightly increasing bulb in the first part of the spectrum that exists in the kidney dataset. In Figure 2.3, the four parameters for the baseline function (2.4) have been plotted at the respective pixel locations. Spectra close to the border of the sample have parameters that lead to a much higher baseline in the beginning and decrease rapidly, while central spectra have a much more constant baseline. The general trend reveals a strong relation of the initial slope of the baseline with the spatial position within the sample, which is also observed by Norris et al. (2007) and House et al. (2011).

### 2.3.1 Mixture based baseline estimation

Andrade and Manolakos (2003) introduced mixture based baseline estimation to replace the naive approach of quantile clipping. The algorithm is available in MATLAB's bioinformatics toolbox. The model assumption is that the intensity of a spectrum is the mixture of an intensity distributed around the baseline and the actual signal. Both distributions are assumed to be disturbed by Gaussian noise. Directly accessing the class or the mixture coefficient is not possible. Therefore, the mass spectrum is divided into several windows just as done for quantile clipping (Williams et al., 2005). In each of the windows, an estimate for the height of the signal background needs to be found. Likewise, once the estimate for the height is found, a regression through these points is made.

Consider one has observed the intensities $x_j,\ j = 1,...,m$ in a window of the mass spectrum. The intensity $x_j$ maps to the hidden class variable $y_j \in \{0,1\}$ coding if $x_j$ belongs to the background signal or to a peak. The probabilistic model of this approach has the simple form

$$P(x_j \mid \theta) = \sum_{c=0}^{1} p_c f(x_j; \mu_c, \sigma_c)$$

with the parameter vector $\theta = (p_0, \mu_0, \sigma_0, p_1, \mu_1, \sigma_1)$. In this model $p_c = P(y_j = c) \in [0,1]$ is the mixing probability fulfilling $p_0 + p_1 = 1$, and $f(x_j; \mu_c, \sigma_c)$ describes the normal density (2.3). Reformulation into a log-likelihood allows the estimation $\hat{\theta}$ of the parameter vector from the observed data $x_j$ in the window. However, since $y_j$ is unknown, this has to be done with an expectation-maximization (EM) algorithm in two iteration steps (Hastie et al., 2009). Since the spectrum model (2.2) consists of positive values for $f_l$ and $\beta$, peaks are only added to the background, therefore, one can assume $\mu_0 < \mu_1$.

Having obtained the estimates within each window, a regression is performed with a piecewise cubic model. Other types of regression are likewise possible as well.

### 2.3.2 Wavelet-based baseline removal

In wavelet-based baseline removal, a smoothed signal is obtained by filtering with a wavelet filter function using *stationary wavelet transform* (SWT). In summary, a wavelet transform uses a filter function to decompose the input signal into a detail signal and an approximation signal (Hastie et al., 2009). Properties of the filter allow a loss less reconstruction of the input signal with the inverse wavelet filter. A wavelet transform can be applied multiple times as a filter bank

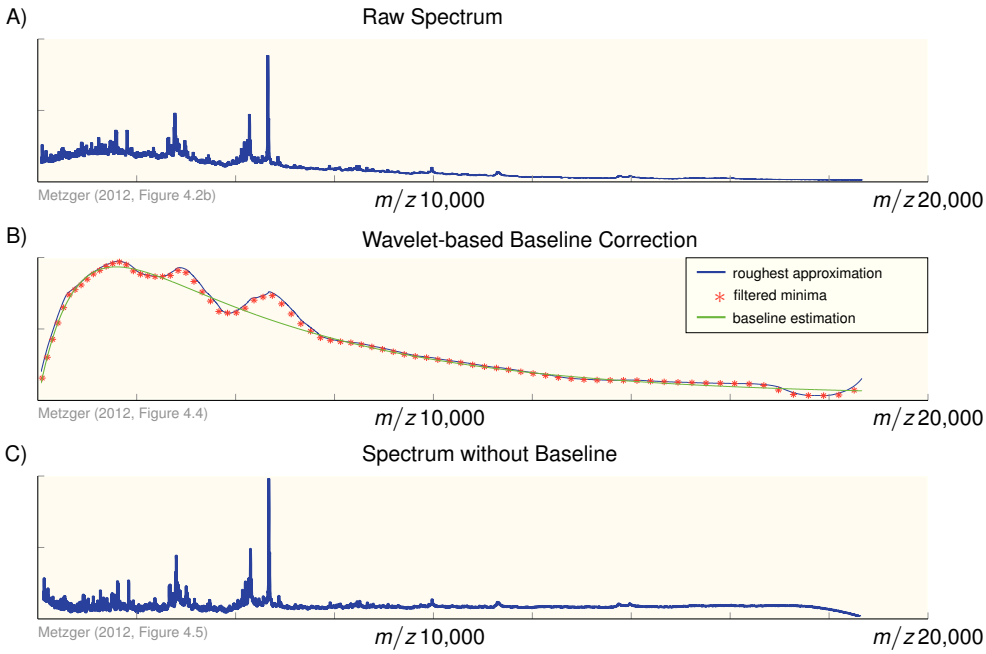



**Figure 2.4: The different stages of wavelet-based baseline correction.** A. Raw spectrum with baseline is approximated by stationary wavelet transform. B. Approximation is interpolated at selected minima by a baseline estimate. C. Wavelet reconstruction of the modified approximation values results in a spectrum with the baseline removed. (Reproduced from Metzger, 2012.)

with $k$ levels. For each level $k$, the wavelet transform is applied to the approximation signal as of level $k-1$. The filter function is a window function that is shifted through the input signal. The traditional approach reduces the length of the signal in each level. This provides a sparse representation of the signal for a known basis of filter functions. In the case of SWT this is slightly different. Instead of a signal reduction, the windows are increased. In general, wavelet transforms have two important applications achieved by thresholding in the wavelet domain. First, suppression of details at the final level $k$ followed by reconstruction, results in a smoothed version of the signal. Second, suppression of coefficients from the approximated signal results in the removal of general trends. In fact, a baseline is such a trend that can be observed in an approximation.

Figure 2.4A displays, a single spectrum from the kidney dataset used as an input signal. On the input signal SWT is performed with $k = 9$ levels to obtain a rough approximation. In this approximation, peaks are removed, as these are captured in the detail function (not displayed in

the figure). The model function that consists of two exponential functions as in (2.4) is then fitted to the approximated signal which are both shown together in Figure 2.4B. To remove the baseline as the trend, all values below the fit model function are set to zero. The result serves as the new approximation for reconstruction with the inverse wavelet filter to obtain the cleaned signal. The information about peaks is recovered from the detail coefficients of the signal. Therefore, only the influence of the baseline is removed, see Figure 2.4C.

Shin et al. (2010) state the clear advantage of the method, is its independence from minimal mass shifts often found in IMS. Independence from mass shifts is a direct result of using the SWT. Moreover, wavelet methods can be efficiently implemented in a graphics processing unit (GPU) framework. Another advantage is the innovative approach to remove the baseline in the wavelet domain. This way adaptive thresholding of the different noise variance is very effective. For more theory about wavelets as well as the application to MALDI data, see the bachelor thesis by Metzger (2012).

## 2.4 Peak picking in mass spectra

With an initial model of peaks in mass spectra selected, a mathematical method of extracting the peaks can be chosen, such as least-square fitting (Kempka et al., 2004; Lange et al., 2006), wavelet transformation (Morris et al., 2008; Renard et al., 2008) or deconvolution (Broersen, 2009; Alexandrov et al., 2010). In Alexandrov et al. (2010) this algorithm is *orthogonal matching pursuit* (OMP; Denis et al., 2009). Since mining each spectrum in the dataset is computationally expensive and, as explained later, not necessary, only a subset of spectra is considered. Usually, a subset consisting of every 10th spectrum is provides the $m/z$ values from the entire dataset (Alexandrov et al., 2010). For those spectra that are considered for peak picking, OMP tries to match the given peak shape to identify a peak at a certain $m/z$ position. OMP is a greedy deconvolution algorithm where a good fit is determined by the height of the peak and its shape. For the shape, a Gaussian (2.3) with parameter $\sigma$ is used. Due to the theoretical properties of OMP it is also able to detect peaks which are hidden in the noise. This is achieved by the step-by-step deconvolution, where intensities corresponding to a previously identified peaks are removed from the spectrum signal leaving its residuum without the identified peak. This allows the detection of the hidden peaks in the residuum. For more detail, see Denis et al. (2009) and the references therein.

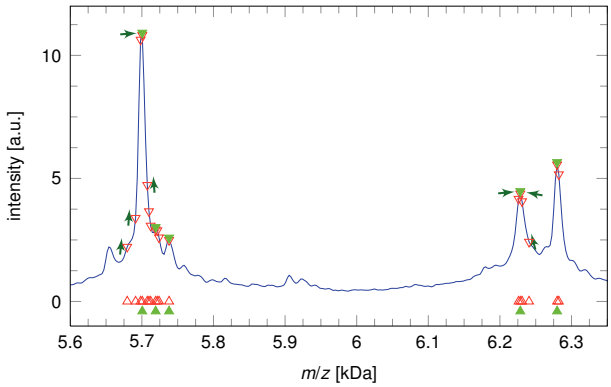



**Figure 2.5: Improved peak picking by alignment to maxima in mean spectrum.** The dataset mean spectrum is shown in blue, Red triangles indicate peaks (and their masses) found after peak picking. Green arrows illustrate the process of alignment. Green triangles show aligned peaks and their masses. (Reproduced from Alexandrov et al., 2011.)

When OMP has found a certain number of $n_p$ peaks in the current spectrum, the algorithm stops. Practically, it would be possible to continue the search for further peaks in a spectrum. However, due to the search being carried out on a large amount of spectra, all peak positions will be considered later on. Therefore, it is not necessary to carry out an exhaustive search on every spectrum. Peaks that are masked in a single spectrum will most certainly be found in other spectra if they are important. On the other hand, it would be impractically to consider all possible peak positions.

After the subset considered for peak picking has been searched, the list of peak positions is refined. First, only the most frequently picked peaks will be considered. Which are those found at least $\tau_p\%$ times in the whole dataset. This way false positives are removed from the list of potential peaks. Second, the remaining $m/z$ values are aligned to the peak maximum found in the mean spectrum of the dataset. Since small mass shifts occur naturally and noise is present in the data, several $m/z$ values identified by OMP account for the same peak in the data. If considering a range of consecutive measured $m/z$ values, this gives the series of values a higher weight in the later processing. Therefore, the redundancy has to be removed in order to achieve good results (Alexandrov and Kobarg, 2011). As already mentioned, a way to remove the redundancy is to move the peak position uphill into the direction of the local maximum found in the mean spectrum. This grouping of peaks is illustrated in Figure 2.5 and has increased the sensitivity of

peak picking without a drop of the specificity (Alexandrov and Kobarg, 2011; Alexandrov et al., 2011).

To summarize, this peak picking algorithm has three parameters, $\sigma$ the standard deviation of the Gaussian shape of a peak, $n_p$ the number of peaks selected per spectrum, and $\tau_p$ the percentage of spectra a peak should be found in and the outcome is a list of $p$ peaks (Alexandrov and Kobarg, 2011). Shapes other than the Gaussian one will be the focus in Section 3.2 and the direct assessment of complete prototype spectra is presented in Section 4.3. However, at this point, the mass spectra measured with $m$ bins are now reduced to a peak list. In the following, the term mass spectrum will still be used for such a signal, but will be denoted the size as $p$ only.

## 2.5 Clustering of multivariate data

As a tool for unsupervised analysis of multivariate data, clustering techniques have been proven to create results that enable fast interpretation of the data (Hastie et al., 2009).

Using hierarchical clustering proved to be very suitable for analysis of mass spectrometry data Schwartz et al. (2004) and in IMS (Chaurand et al., 2004; McCombie et al., 2005; Bonnel et al., 2011), becazse the grouping reveals further information. However, cases of hierarchical clustering failing to provide reasonable segmentation where individual spectra form isolated clusters have been reported (McCombie et al., 2005; Alexandrov and Kobarg, 2011). In contrast, the fast and efficient $K$-means algorithm provides similar results, but lacks the hierarchical information, as the number of clusters must be specified in advance (McCombie et al., 2005). The advantages of both approaches can be combined using bisecting $K$-means (Steinbach et al., 2000) when applied to IMS data (Trede et al., 2012b). For the size of recent IMS data alternative clustering algorithms could be considered, such as CURE (Guha et al., 2001), HDDC (Bouveyron et al., 2007), minimum entropy clustering (Li et al., 2004), CLARANS (Ng and Han, 2002), or BIRCH (Zhang et al., 1997). None of these was developed specifically for IMS and as such also suffer from the same issue of being not suited for the number of spectra and peaks.

Mathematically speaking, a clustering algorithm creates a segmentation $\Xi = \{C_1, \ldots, C_K\}$ of the multivariate dataset $\mathcal{X} = \{x_i \in \mathbb{R}^p \mid i = 1, \ldots, n\}$ with $p$ features into $K$ disjoint subsets $C_k \subset \mathcal{X}$ with

$$\mathcal{X} = \bigcup_{k=1,\ldots,K} C_k$$

such that $C_k \cap C_{k'} = \varnothing$ for $k \neq k'$. Instead of representing the data by the subsets $C_k$ the obtained segmentation, one can also express the result of the clustering algorithm as a vector of labels $y \in \{1,\ldots,K\}^n$, where $y_i = k$ if and only if $x_i \in C_k$.

The segmentation of the dataset $\mathcal{X}$ into subsets with similar features requires the definition of a measure of similarity. In the general case, the similarity is expressed by the dissimilarity function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. For the majority of applications, as in the case of mass spectrometry data, a metric can be used as the dissimilarity function. Common metric functions are the well known Euclidean distance

$$d_2(x_i,x_j) = \|x_i - x_j\|_2^2 = \sum_{l=1}^{p} (x_{il} - x_{jl})^2$$

or the city block distance

$$d_1(x_i,x_j) = \|x_i - x_j\|_1 = \sum_{l=1}^{p} |x_{il} - x_{jl}|$$

for any two objects $x_i, x_j \in \mathcal{X}$. When the angle between observations is more interesting than the actual intensity in each dimension, the correlation coefficient

$$\rho(x_i,x_j) = \frac{\sum_{l=1}^{p}(x_{il} - \bar{x}_i)(x_{jl} - \bar{x}_j)}{\sqrt{\sum_{l=1}^{p}(x_{il} - \bar{x}_i)^2}\sqrt{\sum_{l=1}^{p}(x_{jl} - \bar{x}_j)^2}} \tag{2.5}$$

is a proper choice for dissimilarity. Here $\bar{x}_i = \frac{1}{p}\sum_{l=1}^{p} x_{il}$ denotes an observations mean. As the correlation coefficient is a function $\rho : \mathcal{X} \times \mathcal{X} \to [-1,1]$ it has to be transformed into a similarity with $d_\rho(x_i,x_j) = 1 - \rho(x_i,x_j)$. In the case of centralized data, which means $\bar{x} = 0$, the correlation coefficient (2.5) even reduces to the much simpler form

$$\gamma(x_i,x_j) = \frac{\langle x_i \mid x_j \rangle}{\|x_i\|_2 \|x_j\|_2}$$

with $\langle x_i \mid x_j \rangle = \sum_{l=1}^{p} x_{il} x_{jl}$ being the scalar product. The latter two distance measures work very well in the context of mass spectrometry (Deininger et al., 2008). The reason for this is that the similarity between patterns of intensity for certain masses are more important than actual intensity. The popularity of using distances for the dissimilarity function, leads to the two terms being used synonymous in the context of clustering. Furthermore, several more (dis-)similarity

measures can be considered, such as those from information theory or text mining (Sebastiani, 2002).

For convenience, the pairwise distances can be stored in a distance matrix $D = (d(x_i, x_j))_{ij}$, $i, j = 1, \ldots, n$. Even though, the metric distances are symmetric such that $D = D'$, there are still $n(n-1)/2$ entries from the upper triangle needed to be stored (Kaufman and Rousseeuw, 1990). Since this is depending on the number of observations squared, the distance matrix is considerably large. This poses a problem and demands a careful choice of clustering algorithm (Deininger et al., 2008).

### 2.5.1 Hierarchical clustering with pairwise linkage

When speaking of clustering, two approaches can be considered, hierarchical or partitioning. First, hierarchical clustering will be considered, which again can be divided into two methods, agglomerative and divisive (Kaufman and Rousseeuw, 1990). However, it is often automatically assumed that an agglomerative method is employed. The agglomeration is created by linking the two most similar objects or clusters in a dataset (Kaufman and Rousseeuw, 1990). Therefore it is also known as a bottom-up method. While the similarity between objects can be based on one of the distance measures introduced before, a concept of similarity between clusters requires the definition of a linkage function. The linkage function $d : P(\mathcal{X}) \times P(\mathcal{X}) \to \mathbb{R}$ specifies the dissimilarity in a more general concept. Several types of linkage do exist and will be discussed. Usually it is not possible to specify a generally working linkage and distance for all applications. Based on data and underlying theory the appropriate linkage has to be selected.

Given two non-empty sets $\mathcal{A}, \mathcal{B} \subset \mathcal{X}$ with $\mathcal{A} \cap \mathcal{B} = \varnothing$, the distance

$$d(\mathcal{A}, \mathcal{B}) = \min_{a \in \mathcal{A}, b \in \mathcal{B}} d(a, b)$$

by *single linkage* between the two segments is defined as the minimal distance between all pairs from the sets. Therefore single linkage is also known by the alternative name of *minimal linkage*. Naturally the quite opposite is the *maximal linkage* also called *complete linkage*. Here one defines the distance between the sets

$$d(\mathcal{A}, \mathcal{B}) = \max_{a \in \mathcal{A}, b \in \mathcal{B}} d(a, b)$$

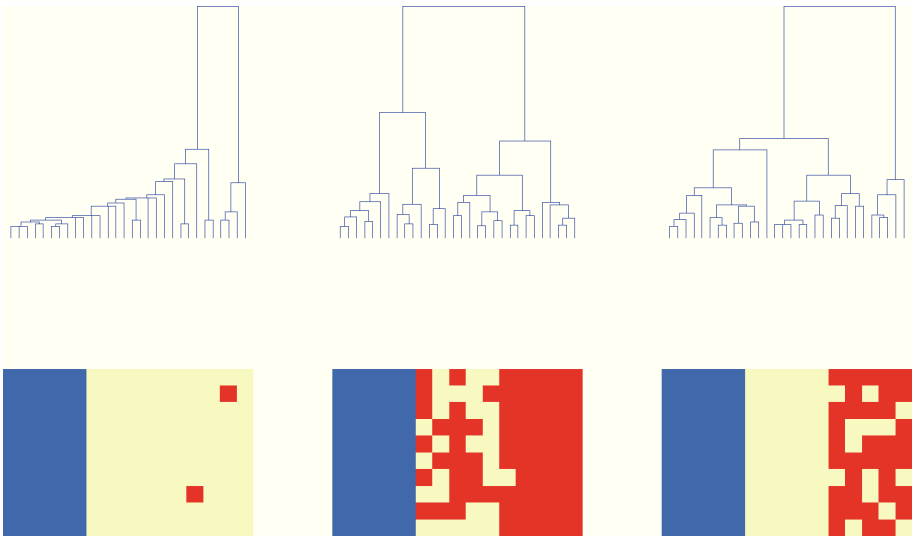


**Figure 2.6: Dendrogram and segmentation map of Fisher's iris dataset for hierarchical clustering with linkage.** From left to right: single linkage, complete linkage, average linkage.

as the pairwise maximum. Aside from these two linkages, the *average linkage* is used

$$d(\mathcal{A},\mathcal{B}) = \frac{1}{|\mathcal{A}||\mathcal{B}|} \sum_{a\in\mathcal{A}} \sum_{b\in\mathcal{B}} d(a,b)$$

to determine the average dissimilarity between all objects.

In all three cases the clustering is initialized by considering all $n$ elements from the dataset as clusters $C_i = \{x_i\}$, $i = 1,\ldots,n$ and the calculation of the corresponding distance matrix $D$. Since the segmentations only contain one element, they are also called singletons. Those two clusters $\mathcal{A},\mathcal{B}$ form a new cluster, where $d(\mathcal{A},\mathcal{B}) = \min_{i,j} D_{ij}$. The linking of clusters is repeated until only one cluster remains such that $\Xi = \{\mathcal{X}\}$ (Kaufman and Rousseeuw, 1990). A so called mergelist keeps track of all linkage actions and contains the pair that form the nested hierarchy.

Exemplary, the clustering results of the three linkage types are visualized for the iris data in Figure 2.6. The most common visualization for the hierarchy is by usage of a dendrogram (Hastie et al., 2009). In contrast to a simple binary tree where just the splits of classes are visualized, a dendrogram reveals the impact of the split. This is achieved by using the distance between the joined clusters as the length of the branch.

Computationally these linkage functions can be implemented quite easily once the distance matrix has been calculated. This is mainly due to the fact that the linkage can be calculated recursively (Lance and Williams, 1967; Kaufman and Rousseeuw, 1990). Assume the clusters $\mathcal{A}$ and $\mathcal{B}$ are to be joined to form the cluster $\mathcal{R}$ with average linkage. Then it is necessary to obtain the dissimilarity $d(\mathcal{R},\mathcal{Q})$ of $\mathcal{R}$ to any other cluster $\mathcal{Q}$ by

$$\begin{aligned} d(\mathcal{R},\mathcal{Q}) &= \frac{1}{|\mathcal{R}||\mathcal{Q}|}\sum_{\substack{r\in\mathcal{R}\\ q\in\mathcal{Q}}} d(r,q) \\ &= \frac{1}{|\mathcal{R}||\mathcal{Q}|}\sum_{\substack{r\in\mathcal{A}\\ q\in\mathcal{Q}}} d(r,q) + \frac{1}{|\mathcal{R}||\mathcal{Q}|}\sum_{\substack{r\in\mathcal{B}\\ q\in\mathcal{Q}}} d(r,q) \\ &= \frac{|\mathcal{A}|}{|\mathcal{R}|}\left(\frac{1}{|\mathcal{A}||\mathcal{Q}|}\sum_{\substack{r\in\mathcal{A}\\ q\in\mathcal{Q}}} d(r,q)\right) + \frac{|\mathcal{B}|}{|\mathcal{R}|}\left(\frac{1}{|\mathcal{B}||\mathcal{Q}|}\sum_{\substack{r\in\mathcal{B}\\ q\in\mathcal{Q}}} d(r,q)\right) \end{aligned}$$

effectively this means the distance

$$d(\mathcal{R},\mathcal{Q}) = \frac{|\mathcal{A}|}{|\mathcal{R}|}d(\mathcal{A},\mathcal{Q}) + \frac{|\mathcal{B}|}{|\mathcal{R}|}d(\mathcal{B},\mathcal{Q})$$

is directly accessible from the information existing before the merge. However for recursive calculation all pairwise distances need to be present in the memory. Effectively, this prevents the application of linkage clustering to huge mass spectrometry datasets (Trede et al., 2012b) and alternatives have to be used.

### 2.5.2 The *K*-means algorithm

Most often the $K$-means algorithm is employed as an alternative to the iterative merging of a dataset based on the defined similarity measure. MacQueen (1967) initially described $K$-means with the objective to find from all possible segmentations $P(\mathcal{X})$ the segmentation with minimal variance within the clusters. Since then, it had multiple applications and is used for mass spectrometry data mainly due to its high efficiency on large datasets (Alexandrov and Kobarg, 2011). In contrast to the hierarchical method from the previous section $K$-means is a partitioning clustering where the number of segments has to be specified in advance. However, $K$-means can be extended to a hierarchical variant as well, which is introduced in the next section.

At first, the algorithm is initialized with centroids $\xi_1,\ldots,\xi_K \in \mathcal{X}$ to generate clusters. The centroids are selected randomly or by a selection strategy discussed later. For each of the selected

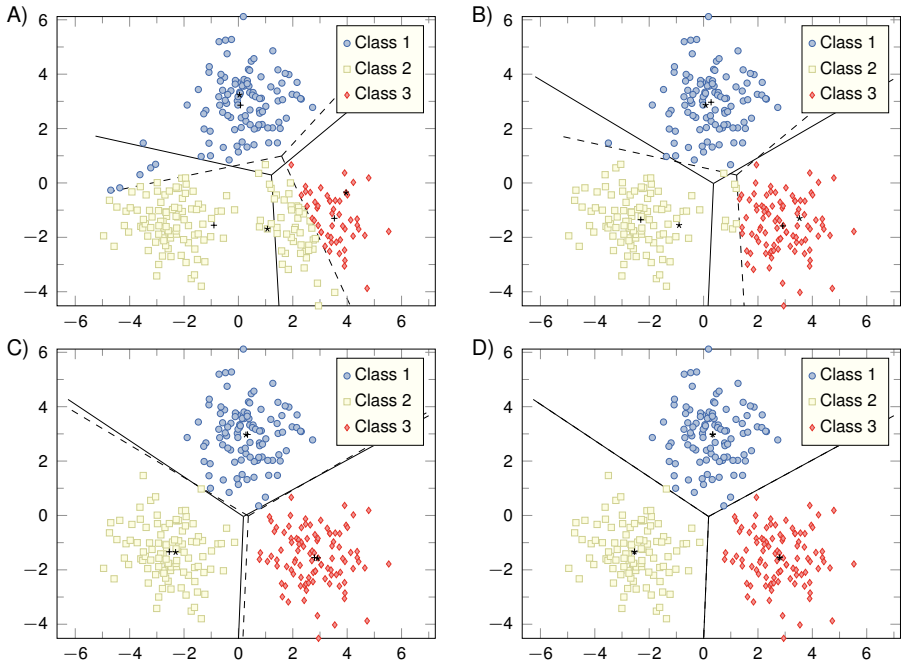



**Figure 2.7: Individual steps in K-means clustering.** Each image displays the segmentation based on centroids after $n = 1,...,4$ iterations. The dashed line shows the decision boundary before and the solid line after the corresponding iteration.

centroids the distances to the objects $x \in \mathcal{X}$ are calculated. So instead of $n(n-1)/2$ entries of a distance matrix, only $nK$ distances are needed. The objects $x \in \mathcal{X}$ are assigned to the cluster

$$C_k = \{x \in \mathcal{X} \mid k = \underset{\kappa=1,...,K}{\arg\min}\, d(\xi_\kappa, x)\}$$

generated by the closest centroid $\xi_k$. The next step will update the centroids as the mean of all objects

$$\xi_k = \frac{1}{|C_k|} \sum_{x \in C_k} x$$

in each cluster, where $|C_k|$ denotes the number of elements in $C_k$. The centroid update is shown in Figure 2.7 and is repeated until no element is reassigned.

A problem that has to be faced is the fact that the solution found by $K$-means is only a local solution (Steinley and Brusco, 2007). Based on the initialization different cluster centroids can be found and therefore different cluster maps are obtained. Implementation such as the one

from MATLAB solve this problem by using different initializations and declaring the one as representative which has the most compact clusters. The clusters are named compact, if the variance within the obtained groups is small (Mirkin, 2005). Alternative strategies involve the selection of elements that form a convex hull of the data (Mirkin, 2005; Chiang and Mirkin, 2010). For example, by selecting $\xi_1 = \arg\max_{x \in X} d(x, \bar{x})$ and then

$$\xi_k = \arg\max_{x \in X} \sum_{\kappa=1}^{k-1} d(x, \xi_\kappa)$$

for $k = 2, \ldots, K$ poses one deterministic selection strategy. For a complete review of alternative methods, see Steinley and Brusco (2007).

### 2.5.3 Bisecting *K*-means

While the speed and low memory need of $K$-means are major advantages of the algorithm in contrast to hierarchical clustering with a linkage, the drawback is the need to specify a desired number of clusters in advance. Often, this is not known and several numbers have to be tried. However, due to the fact that $K$-means tries to create an optimal segmentation into $K$ segments, objects will not necessarily be grouped together across the results. So for a plausible segmentation into $K$ classes, one might obtain an implausible result for $K+1$ classes. This is most often the case, as the $K+1$ classes are need to be found together rather after each other.

Fortunately, a simple modification of $K$-means will avert this disadvantage by using the bisecting approach proposed by Steinbach et al. (2000). Instead of directly searching for an optimal segmentation, each iteration only searches for a split into exactly two classes. The process is repeated for each of the subsets until each segmentation only contains singletons. Since one starts with the full data and end with singletons, this is also called top-down clustering. Earlier top-down clustering by divisive analysis has been proposed (Kaufman and Rousseeuw, 1990). Kaufman and Rousseeuw (1990) propose to iterate through all elements to find a sequence of elements that best form their own cluster. This approach is sequential like agglomerative clustering and does not scale well to large data. It has not gained the same popularity as agglomerative hierarchical clustering or $K$-means (Hastie et al., 2009).

Computationally, bisecting $K$-means has a fast run time, as only a two-fold segmentation needs to be obtained and the cardinality of the sets to work on decreases with each iteration. The problem of switching labels is avoided by this approach. Furthermore, it re-introduces the

| | estimate | |
|---|---|---|
| known state | positive $\hat{y} = 1$ | negative $\hat{y} = 0$ |
| true $y = 1$ | true positive | false negative |
| false $y = 0$ | false positive | true negative |

**Table 2.1: Confusion matrix for classification evaluation.** Confusion matrix showing the relation between a known state and the positive or negative outcome of an estimator.

possibility to display the hierarchical splitting of the data. With this method, segmentation of very large datasets can be achieved (Trede et al., 2012b; Oetjen et al., 2013).

### 2.5.4 Evaluation measures for segmentation results

A difficult question to answer is how to evaluate the agreement between the segmentation maps that are obtained when initialization parameters are changed or when the best choice of clustering is not known. The question becomes easier with actual labels known, as it is the case in training a classifier with (semi-) supervised statistical learning (Hastie et al., 2009). Training of a classifier is a task in supervised data analysis where the goal is to obtain a label for new data based on previously learned model from training data. Even though, training of a classifier is different from segmentation, here the evaluation measures are introduced to compare segmentation maps from clustering. Recall the segmentation results obtained by hierarchical clustering in Figure 2.6. In this case, the actual label each object has is known and the quality of the segmentation can be evaluated. Manual alignment of the class label and the clustering result has to be performed. As a starting example, evaluation of the complete linkage result for *I. versicolor* (yellow) and *I. virginica* (red) will be done. In total 28 flowers were assigned to *I. versicolor* and 72 to *I. virginica*. Obviously, several flowers were misclassified as there are only 50 of each species. In detail, complete linkage assigned 23 flowers to the species *I. virginica* while it is truly *I. versicolor*, yet only one of *I. virginica* is incorrectly identified as *I. versicolor*.

One family of suitable quality measures can be obtained from a *confusion matrix* (Sebastiani, 2002). First consider the simple case of comparing label estimation for two classes. Then one has for $n$ observations the known labels $y = \{0,1\}^n$ and an estimate $\hat{y} = \{0,1\}^n$. To distinguish between the labels and estimates, one usually speaks of classes true and false for the known labels (for example health status) and of positive and negative for the estimates (in the sense of identifying the health status). In this setting, a confusion matrix is a two-by-two matrix as shown

in Table 2.1. The entries of the matrix count all those observations that belong to the true class and are classified correctly as such (true positive, $TP = \#\{y \wedge \hat{y}\}$) and those that are correctly classified to the false class (true negative, $TN = \#\{\neg y \wedge \neg \hat{y}\}$). Naturally both false negatives ($FN = \#\{y \wedge \neg \hat{y}\}$) and false positives ($FP = \#\{\neg y \wedge \hat{y}\}$) exist. From these four values a measure of classification quality can be obtained. In practice the last two values should reduce the score as misclassification occurs. Therefore, good scores are the *true positive rate*

$$TPR = \frac{TP}{TP + FN} \tag{2.6}$$

also called *sensitivity* or *recall* and the *true negative rate*

$$TNR = \frac{TN}{TN + FP} \tag{2.7}$$

also known as *specificity* (Hastie et al., 2009). Depending on the perspective, either of the scores can be controlled to be maximized or minimized. In our example, the true positive rate for *I. versicolor* is $\frac{27}{50} \approx 0.54$ and the true negative rate $\frac{49}{50} \approx 0.98$.

The perspective has an influence on the interpretation: If the *I. virginica* species were posing a threat, just one of the threats was missed, so the obtained segmentation can be considered a good result. However, by looking at just one of the scores, trivial classification can occur by assigning all observation only positive or negative labels. Furthermore, if the two examined classes are equally valid, one is usually interested in the overall performance. Therefore, a compromise is to maximize the *balanced accuracy* which is the arithmetic mean of the two error rates (Sebastiani, 2002). Instead of the arithmetic mean, the *F-Score*

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

as the harmonic mean of recall (2.6) and *precision*

$$PRC = \frac{TP}{TP + FP} \tag{2.8}$$

is also a popular measure (Hastie et al., 2009). The precision of *I. versicolor* is $\frac{27}{28} \approx 0.96$, listed here for completeness.

In general, the above mentioned approaches are only valid if one really knows the underlying labels. Especially in our example, the complete linkage only produces labels blue, yellow,

and red. These methods are therefore only applicable in the supervised or semi-supervised context. Great care is advised when they are used to access the similarity between unsupervised segmentation maps without adjusting for chance permutations of the labels.

### Expansion to multi class

In the case of $K > 2$ labels $y = \{1, \ldots, K\}^n$, the notation changes slightly, but the concept can be transferred. Consider the case that the labels are ordered and the estimator $\hat{y}$ can produce the class assignments in the corresponding manner. Alternatively, a supervised correction for the labels is needed. Then the confusion matrix is a $K \times K$ matrix $T = (t_{ij})$, where each entry of $T$ is defined as $t_{ij} = \#\{y = i \wedge \hat{y} = j\}$. For the multi class confusion matrix

$$TP_k = t_{kk} \quad \text{as true positives,} \quad FN_k = \sum_{j \neq k} t_{kj} \quad \text{as false negatives,}$$

$$FP_k = \sum_{i \neq k} t_{ik} \quad \text{as false positives,} \quad TN_k = \sum_{i \neq k} \sum_{j \neq k} t_{ij} \quad \text{as the true negatives}$$

are defined for each of the possible labels $k = 1, \ldots, K$, and one obtains $K$ confusion matrices containing $TP_k$, $TN_k$, $FP_k$, $FN_k$. From these the global confusion matrix is computed, which simply sums up all individual entries, for example $TP = \sum_{k=1}^{K} TP_k$. Using the values from the global confusion matrix allows computation of the scores sensitivity (2.6), specificity (2.7), and precision (2.8) as before (Sebastiani, 2002).

One can also consider to do the computation of the local scores like $TNR_k = \frac{TN_k}{TN_k + FP_k}$ and then generate $TNR = \sum_{k=1}^{K} TNR_k$. However, this can produce a different result, especially when there are categories with few observations, that have little influence otherwise. For a detailed discussion, see Sebastiani (2002). As in the simple case, again one faces the problem that the segmentation result has to match each others labels.

### Other quality methods for segmentation

A different, very well known measure of quality is the so called *Rand index* which was first proposed by Rand (1971) and is reviewed by Hubert and Arabie (1985). The Rand index will compare object pairs and how they are grouped by two different segmentations $\Xi_1, \Xi_2$ putting the focus on consensus among the segmentation maps. Let $n_{00}$ denote the number of objects placed both times in the same class and let $n_{11}$ be the number of objects assigned both times to different

classes. Since the objects itself are used in pairs, no need for label alignment is necessary, but this way, no judgement is obtained. The Rand index

$$R(\Xi_1, \Xi_2) = \frac{n_{00} + n_{11}}{\binom{n}{2}} \tag{2.9}$$

is defined by the agreement between two segmentations (Rand, 1971). The agreement is divided by the total number of possible pairs. Hubert and Arabie (1985) give alternative divisors to adjust for chance grouping or to take the disagreement into account as well. Details about the alternative corrections are found in their publication, while in this thesis an efficient calculation will be explored.

Directly related to the Rand index is the *matching coefficient* by Ben-Hur et al. (2001). For this the *consensus matrix* $M \in \{0,1\}^{n \times n}$ of a segmentation $\Xi$ with entries

$$M_{ij} = \begin{cases} 1, & x_i \text{ and } x_j \text{ belong to the same cluster and } i \neq j, \\ 0, & \text{otherwise,} \end{cases} \tag{2.10}$$

has to be constructed. The consensus matrix is also called connectivity matrix and is practical in the sense that it groups the objects from the dataset based on their paired appearance in the clusters as needed for the Rand index. This makes two different segmentation sets comparable, as the arbitrarily chosen labels are ignored. In Figure 2.8 two matrices are shown for Fisher's iris data. For the ground truth with known labels blocks of object pairs can be seen. The result of complete linkage has a different appearance in the consensus matrix. The upper left block is the same, but from this visualization one can conclude that the result for the other two classes is inconclusive. Since the order of the objects was preserved, the block structure will only appear when the rows and columns are re-ordered, such that they are sorted by class label.

The use of the consensus matrix (2.10) allows the definition of

$$\langle M^{(1)} \mid M^{(2)} \rangle = \sum_{i,j} M_{ij}^{(1)} M_{ij}^{(2)} \tag{2.11}$$

as the dot product between two consensus matrices (Ben-Hur et al., 2001). Since this dot product will only count entries being zero or one, the value can be computed by boolean operations. Even

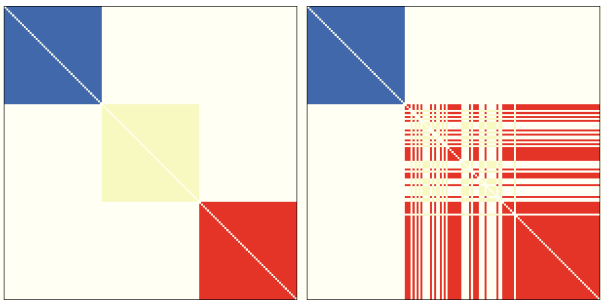


**Figure 2.8: Consensus matrices for Fisher's iris data.** Two consensus matrices, consensus matrix of the known labels (left) and consensus matrix of complete linkage (right). The class colour is only added to aid getting the correspondence between the segmentation images presented earlier.

for large matrices these operations are fast and require few memory. Therefore, the number of comparing pairs

$$n_{11} = \langle M^{(1)} \mid M^{(2)} \rangle \qquad n_{01} = \langle \neg M^{(1)} \mid M^{(2)} \rangle$$
$$n_{10} = \langle M^{(1)} \mid \neg M^{(2)} \rangle \qquad n_{00} = \langle \neg M^{(1)} \mid \neg M^{(2)} \rangle$$

are defined using the dot product (2.11) and one can obtain the values needed for computation of the Rand index (2.9).

Beside using the consensus matrix for efficient calculation of the Rand index, it can also be used to obtain a grand consensus of more than one segmentation map (Monti et al., 2003). The sum of several consensus matrices $M^{(1)}, \ldots, M^{(r)}$ divided by their number transforms them into a dissimilarity matrix for the dataset. This dissimilarity matrix can be used as the input for the hierarchical clustering algorithms from Section 2.5.1. This approach can be used to remove the influence of outliers when each segmentation is only performed on a (random) subset.

## 2.6 Accessing mass-to-charge values with segmentation maps

After the spectra have been grouped by a segmentation algorithm, the search for important $m/z$ values is the next step. Usually, segmentation is performed on peak picked spectra $x_i \in \mathbb{R}^p$ for performance reasons. Therefore, the search for important $m/z$ values has to consider the full spectra $x_i \in \mathbb{R}^m$, $i = 1, \ldots, n$, again. The labels $y$ of the segmentation map will create $k = 1, \ldots, K$ spatial masks $b^k \in \{0,1\}^n$ with $b_i^k = 1$ if and only if $y_i = k$. Van de Plas et al. (2007) and

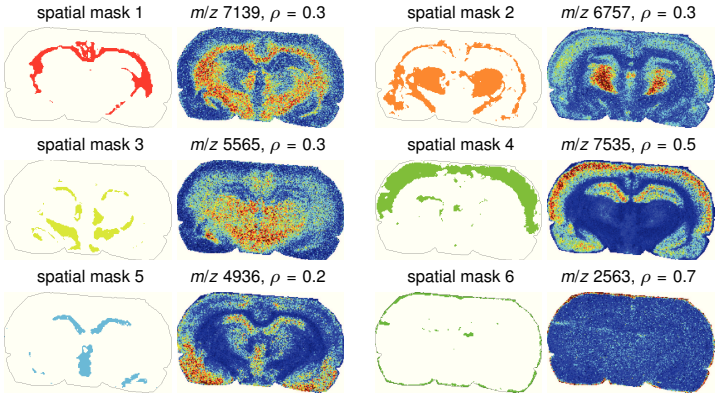



**Figure 2.9: Pearson correlation chart for rat brain dataset.** For each of the segments the $m/z$ image with the highest correlation is shown. (Reproduced from Alexandrov et al., 2010.)

McDonnell et al. (2008) have proposed to use the correlation coefficient (2.5) to find $m/z$ values at which the intensity is co-localized with a spatial mask. In Figure 2.9 this principle is shown for the six spatial masks obtained by Alexandrov et al. (2010). With this approach it is easy to gain a list of values that are important for the found segments (Bruand et al., 2011b; Suits et al., 2013). Out of the full list of $m/z$ values, the one with the highest correlation is displayed in the figure. Naturally, the correlation coefficient can also be calculated between any two given $m/z$ values from the dataset. For example, a $m/z$ value relating to a known compound can be examined with the correlation to search for other co-localized $m/z$ values. In theory, these $m/z$ values can be fragments or reveal anti correlation.

Evidently, the correlation coefficient is a simple way to access the spatial location of $m/z$ values. However, technically it relies on the data being provided in the nature of images rather than spectra. This is often not the case from the data storage perspective, as spectra are provided pixel by pixel. The reduction of spectra to peak lists is not only natural from the biological side of application, but also practically from a computation and memory storage side. Moreover, naive reading of the image data as chunks into the available memory slows down the whole process. Improper setting of chunk sizes causes multiple read times from file. However, Pearson's correlation formula (2.5) consists of several sums, therefore it is possible to load a single spectrum, store the summands for each $m/z$ channel and compute the correlation values after each spectrum has been loaded.

Mathematically speaking, the correlation coefficient between a bool mask $b \in \{0,1\}^n$ for spectra $x_i \in \mathbb{R}^m$, $i = 1,\ldots,n$, in $j$-th $m/z$ channel $x^j \in \mathbb{R}^n$ is

$$r(x^j, b) = \frac{\sum_{i=1}^{n}(x_{ij} - \bar{x}^j)(b_i - \bar{b})}{(n-1)\hat{\sigma}_{x^j}\hat{\sigma}_b} \tag{2.12}$$

which is a slight reformulation of (2.5) with respect to the number of spectra $n$. In order to change from image wise computation to spectrum wise, first one expands and then reorders the sums

$$\frac{n\sum_i x_{ij}b_i - \sum_i x_{ij}\sum_i b_i}{\sqrt{n\sum_i x_{ij}^2 - (\sum_i x_{ij})^2} \cdot \sqrt{n\sum_i b_i^2 - (\sum_i b_i)^2}} = \frac{nS_0^j(b) - S_1^j S_b}{\sqrt{nS_2^j - (S_1^j)^2} \cdot \sqrt{nS_b - S_b^2}} \tag{2.13}$$

where in short the sums over spectrum values are replaced by

$$S_0^j(b) = \sum_{i=1}^{n} x_{ij}b_i \qquad S_1^j = \sum_{i=1}^{n} x_{ij} \qquad S_2^j = \sum_{i=1}^{n} x_{ij}^2$$

and a constant $S_b = \sum b_i$ which only depends on the mask.

Naturally, the vectors of sums are obtained by sequential reading of all spectra. Therefore, the vectorized form (2.13) is equal to the correlation (2.12) in the original form. Even more so, it becomes clear that the correlation can be instantly calculated for several different spatial masks $b^k$ during reading. Each spatial mask only requires its own vector of sums $S_0^j(b^k)$.

## 2.7 Summary of processing pipeline

In this chapter, a computational approach to process imaging mass spectrometry data has been given. Rather than the pure application presented in Alexandrov et al. (2010) the individual steps have been examined in more detail with some theoretical background being provided. This allows the formulation of improvements and alternatives in the next chapters.

One aspect of spectrum processing also considered by some groups is the need to align the peak bins (Wolski et al., 2005; Norris et al., 2007). As briefly mentioned in Section 2.4, small mass shifts have to be taken into account. Common workarounds are the binning of data or using methods such as the alignment of peaks to the mean spectrum (Alexandrov and Kobarg, 2011). However, this is a technical issue best left to the software used to acquire the spectra. Likewise, the presented methods for baseline removal are by no means exhaustive. A literature research reveals numerous method proposals (Gibb and Strimmer, 2011).

In the second half of this chapter different clustering methods were explored. The use of hierarchical clustering for IMS was proposed by Chaurand et al. (2004) later data reduction was performed with principal component analysis (PCA) by McCombie et al. (2005), PCA is a method that will be focused on in Section 4.2. Originally, the authors proposed Euclidean distance and with Ward's linkage. However, usage of Ward's linkage is often unattainable due to high memory need for IMS data (Alexandrov and Kobarg, 2011). For the rat brain dataset, average linkage is about 300 times faster than Ward's linkage. Therefore, Ward's linkage was not highlighted as a feasible method and use of correlation with average linkage is recommended (Deininger et al., 2008). Moreover, it was later shown that segmentation algorithms such as $K$-means is even faster due to the top down approach (Trede et al., 2012b).

A new topic introduced here, is the usage of evaluation measures for the comparison of segmentation maps. This helps to objectively access their quality, granted that the ground truth is known. The objective evaluation is essential for determining the usefulness of new methods for processing of mass spectra or to explore the effect that the change of parameters have on the segmentation map. Finally, the connection of segmentation maps for automated segmentation of the data has been drawn to fetch interesting $m/z$ values by correlation analysis. This was the final part of the pipeline by Alexandrov et al. (2010), but required the full data to fit into memory. A small reformulation (2.13) has shown, this is not a restriction.

# 3 Modelling and simulation of MALDI-TOF spectra

## 3.1 Motivation for this chapter

While MALDI-TOF is already a useful tool in various biological applications of analysing 2D tissue sections with imaging mass spectrometry (IMS), it is desirable to surpass being restricted to a 2D sample. Recent improvements of the hardware allowed the acquisition of a whole 3D specimen and digital realignment (Crecelius et al., 2005; Andersson et al., 2008). Research of several sections for a 3D dataset is already routinely possible thanks to standardized preparation protocols and advanced, automated hardware (Andersson et al., 2008; Trede et al., 2012b).

Automated matrix application such as using the ImagePrep device (Bruker Daltonik, Bremen, Germany) or the SunCollect (SunChrom, Friedrichsdorf, Germany) allows fast and standardized preparation of tissue sections and provides the necessary cross-sample quality. This is essential, for merging of sequentially obtained 2D measurements into a single 3D dataset. Standardized preparation with automated steps is the only way to provide high data quality across tissue sections. Without cross-sample quality, features of interest that have a three-dimensional distribution could not be identified. The increasing repetition rate of lasers will allow the acquisition of data to be performed in short time. While now repetition rates of 200–1000 MHz are standard, first results of lasers with rates up to 20 kHz have been reported (Brown et al., 2010; Trim et al., 2010) and are already combined with continuous profiling methods (Spraggins and Caprioli, 2011).

Already 2D IMS delivers large datasets which are considerably difficult to process (Alexandrov et al., 2010). For a sample of 0.5 cm², the number of spectra (pixels) is usually of the size of $n = 20{,}000$ with $200 \times 100$ as the image size at 50 µm spatial resolution. The length of each spectrum poses greater computational problems: a mass spectrum can contain of 10,000 data points, high resolution spectra can contain more than 50,000 data points. Bringing the technique to the third dimension a 3D object is sliced as usual and later the individually measured sections

are reconstructed to obtain the original structure (Crecelius et al., 2005; Trede et al., 2012b). With each section being 10 µm thick (Goodwin et al., 2008), it takes one thousand sections to reconstruct the entire anatomy of the sample under research (Trede et al., 2012b). It is quite acceptable to measure only every fifth section, which saves measurement time and also allows the creation of equidistant grid in which the data points exist.

With increasing acquisition by improved hardware, the algorithms to process the datasets are often slow due to the fact that they mostly depend on the number of obtained spectra. This is because availability of such huge datasets is usually restricted to the labs acquiring them. Even though the sample preparation is standardized, the difficulties and costs of a 3D experiment are high. Therefore, computationally groups have little chance to test their improvements. Furthermore, evaluating IMS data has to be performed in an objective and possibly unbiased manner that requires a gold standard dataset. A simple, common approach makes use of mean spectra obtained by segmentation, followed by replication of the mean with added noise to create arbitrarily sized data (Hanselmann et al., 2008; Kobarg and Alexandrov, 2013; Race et al., 2013). However, this introduces bias, as the underlying data model is already proven to be identifiable.

Instead, a statistically simulated dataset should be used. This allows a researcher to create data with known and adjustable ground truth. Furthermore, a flexible simulation framework allows a to control the level of noise. In this thesis, the solution for simulation of spectra by Coombes et al. (2005) is employed, where the authors propose to model the flight time of particles in a vacuum tube. Another article dealing with simulation of spectra was published by House et al. (2011), where the main emphasis is on finding probability distributions of simulation parameters. However, so far only MS datasets have been created. In this chapter, both approaches will be combined to create simulated IMS and extend the framework to generate a 3D dataset of the anatomy of a mouse brain.

Generation of simulated data can be categorized into replication approaches and simulations. Simulation of mass spectra has been investigated by Coombes et al. (2005); Renard et al. (2008); Schulz-Trieglaff et al. (2008); Bielow et al. (2011); House et al. (2011), and Deininger et al. (2012). The approaches range from complete simulation of peak models to mixing real-life data with a noise model. Like the approaches by Coombes et al. (2005) and House et al. (2011), the methods described in the following survey often do not account for the large number of spectra needed for an imaging dataset. There is also no automatic generation of parameters to generate multiple spectra.

Replication of real-life data was also used by Hanselmann et al. (2008) to create validation datasets for probabilistic latent semantic analysis (PLSA). It comprises three classes for which the authors use the average of real-life SIMS spectra as templates. New spectra are generated by adding Poisson noise to the template spectra. A MALDI imaging dataset was created by Deininger et al. (2012), where for a given peak list the flexControl simulation mode was used. The authors prespecify classes and generate spectra for these. For the final dataset, a low resolution sample is created by the combination of every two-by-two pixel neighbourhood to form a single pixel. For the new pixel, the spectra of its sub-pixels are combined into a mixed spectrum. However, this approach is primarily employed to demonstrate the performance of PLSA as introduced by Hanselmann et al. (2008). The replication performed by Race et al. (2013) uses all spectra available from real-life tissue to create a 3D mouse brain. Noise is added to a smoothed version of the spectrum randomly chosen from the real-life data.

The simulated data used by Kobarg and Alexandrov (2013) also rely on real-life data. However, a small number of distinct spectra were converted into line spectra by peak picking. Noise was added to the line spectra and the time-of-flight model by Coombes et al. (2005) was then used to distort the data. This constitutes an intermediate step between a true simulator and a replication approach.

A complete biological approach is introduced by the LC-MSsim software (Schulz-Trieglaff et al., 2008). The user can specify the proteins and their abundances to be found in the sample. The software then simulates the digestion to generate the spectrum profile of the sample. The hulls of the peak shape are modelled according to the exponential Gaussian hybrid function (3.16). Shot noise, a baseline and impurities can be added to the spectra. The software is available as a module in the OpenMS library. The functionality of LC-MSsim is extended by MSSimulator (Bielow et al., 2011). Most notable is the fact that the latter is also able to model the charge state of electrospray ionization (ESI) or MALDI spectra. This allows to test algorithms on biologically relevant simulated spectra.

Another approach is followed by Renard et al. (2008) who suggest to randomly select digested product peptides and intensities from data base. The generation of isotope patterns is realized by a multinomial distribution. Intensities are convoluted by a Gaussian aperture function dependent on $m/z$ value. The spectra obtained for the specified peptides are combined and Poisson noise is added. It is possible to set the parameter of the Poisson noise level by specifying the signal to noise ratio.

The concept of the here described simulation model will merge the simulation models by Coombes et al. (2005) and House et al. (2011) into a unified model. In the model a spectrum

$$s(t) = \sum_{l=1}^{p} a_l f_l(t) + \beta(t) + \varepsilon(t) \tag{3.1}$$

is a function of the flight time $t$ that consists of the sum of $p$ peak hulls $f_l(t)$, $l = 1, \ldots, p$, and a baseline function $\beta(t)$ with some noise $\varepsilon(t)$. This scheme is identical to all spectra in the simulated dataset. However, the parameters to generate both $f_l$ and $\beta$ of the spectrum at a given spatial position will depend on the statistical simulation the different classes found in the data. Since this approach requires the definition of plausible peak shape functions (House et al., 2011), a review and mathematical representation of established peak shape models is performed (Di Marco and Bombi, 2001).

3D IMS datasets quickly surpass 100 thousand spectra and even consist of several million spectra (Race et al., 2013). Using statistical simulation, generation of these many spectra can be achieved much faster and cheaper than in a laboratory experiment. Possible disadvantages are the assumption of too simple interactions or variability in the data. Most experiments in the context of IMS use organs from rodents, most prominently kidney and brain tissue (Watrous et al., 2011; Seeley and Caprioli, 2012). Therefore, the annotations by Allen Mouse Brain Atlas (2009) are used to statistically simulate a realistic tissue object to be used for the evaluation of new methods.

## 3.2 Modelling of individual mass spectra

In this section, MALDI spectra are explored with two different types of approaches. This has been done in different publications (Di Marco and Bombi, 2001), but was never fully part of the pipeline established by Alexandrov et al. (2010). First, the approach by Coombes et al. (2005) to model the physical process of particles in a vacuum tube is reviewed. Later this model will be used for statistical simulation of a complete dataset. Second, from both the theoretical background and observation of real-life data, one can see that Gaussian peak shapes do not fully model the shape of a peak in a general form. The peak shape has different appearance, depending on the mass. This leads to the conclusion, the peak shape parameters are a function of mass. The second part of this section will compare different peak shape models and evaluate their potential

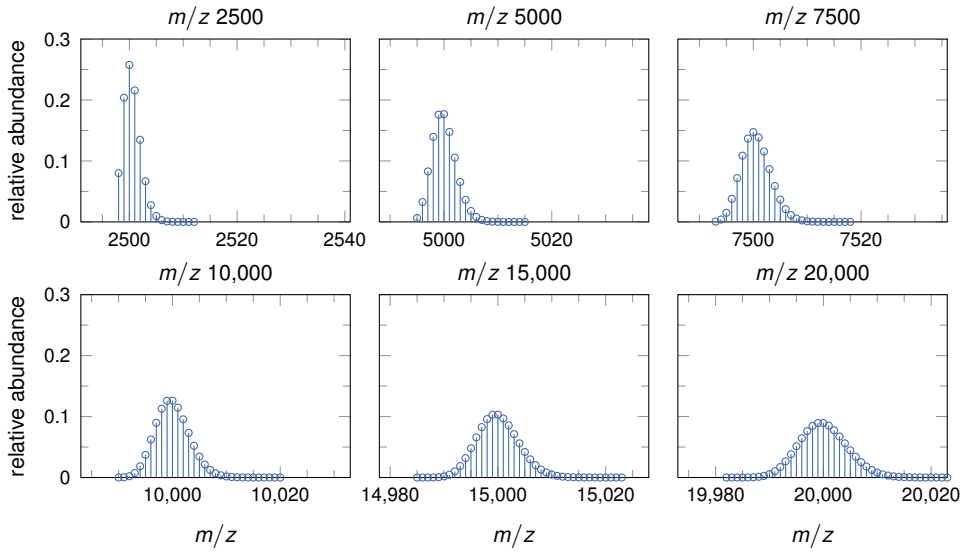



**Figure 3.1: Approximation of isotope distribution.** The probability distribution functions for different nominal masses according to Coombes et al. (2005).

to be fit to data. A way to efficiently compute parameters for the presented peak shape models is also described.

### 3.2.1 Preliminary considerations

Before describing the model from Coombes et al. (2005) in detail or the peak shape functions presented later, two aspects influencing the shape of a peak need to be discussed.

One aspect that has to be considered in mass spectrometry is the so called isotope distribution. In low resolution spectra, like MALDI-TOF, the isotope distribution cannot be observed directly (Russell and Edmondson, 1997; Spengler, 2013). However, since the different masses differ by one unit, the shape of the isotope distribution will largely influence the overall peak shape. Therefore, peak shapes with asymmetry need to be found (Kobarg et al., 2014).

There exist several isotope variants for all atoms. These isotopes have different mass due to the addition of neutrons. When a specific chemical formula is given the isotope distribution can be approximated based on the most abundant atoms for which natural occurrence is known (Kubinyi, 1991; Nicolardi et al., 2010). In the case where the chemical formula is not known an approximation based on the mass has to be made. Coombes et al. (2005) use the binomial

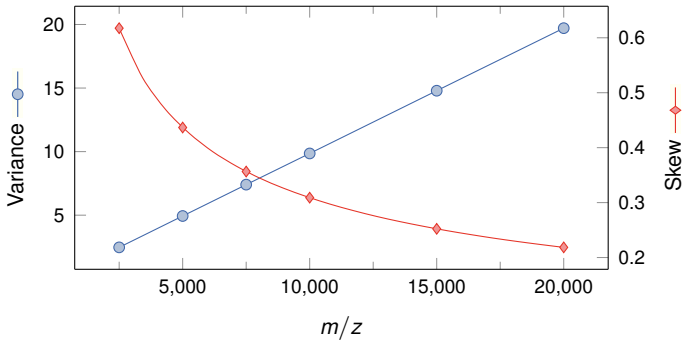



**Figure 3.2: Theoretical values of parameters in peaks.** Parameters describing the isotope distributions depending on mass.

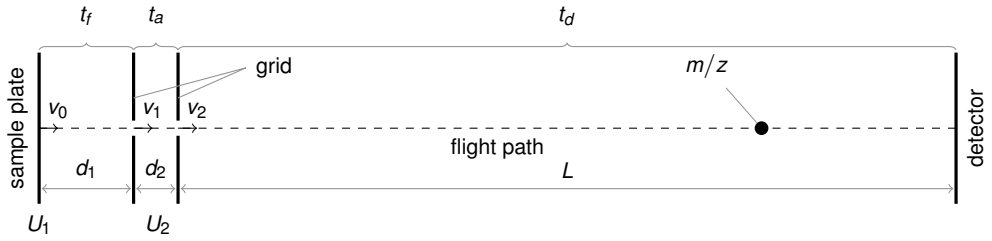
distribution $B_{n,p}(k) = \binom{n}{p} p^k (1-p)^k$ as a model for the mass $n$ with parameter $p = 0.015$. As seen in Figure 3.1 the distribution of the possible isotopes changes from a very asymmetric function for low mass values to an almost symmetric one for large masses.

Another observation that can be made from Figure 3.1 is the increasing variance of the peak shape for increasing mass. Due to the overlap of convoluted peak shapes this results in increasing variance for a unifying peak shape while in fact the measurement uncertainty of the individual isotope peaks stay the same. This effect can be shown in Figure 3.2 where the variance and the skew of the peak shapes from the binomial distribution are plotted against the mass.

This convolution effect is sometimes also confused with a second theoretical aspect. One characteristic of using a flight tube is of course the fact that the mass is only measured indirectly. The observed time $t$ the ions needed to arrive at the detector is converted to $m/z$ value. This is done with the conversion rule by Titulaer et al. (2006) which requires three calibration constants. The quadratic equation

$$0 = A\left(\sqrt{\tfrac{m}{z}}\right)^2 + B\sqrt{\tfrac{m}{z}} + C(t_j) \tag{3.2}$$

describes the relation between $m/z$ and a detector tick $t_j = \delta + \Delta t \cdot j$ with time offset $\delta$ and detector interval $\Delta t$. In practice, a set of calibration masses is used to compute corresponding flight times (Gobom et al., 2002). For the physics based time-of-flight model this conversion is computed accordingly. Then a quadratic function is fit to the observed time-of-flight for the specified instrument parameters (Coombes et al., 2005). The quadratic function (3.2) mapping the detector ticks to $m/z$ values increases the appearance of increasing variance with high masses. For




| Name | Value | Description |
|---|---|---|
| $d_1$ | 17 mm | distance from sample plate to first grid |
| $d_2$ | 8 mm | distance between charged grids |
| $L$ | 1 m | length of the drift tube |
| $U_1$ | 2 kV | voltage used in the ion focusing phase |
| $U_2$ | 20 kV | voltage between charged plates |
| $\delta$ | 600 ns | delay time before focus voltage is applied |
| $\mu_{v_0}$ | 350 m/s | mean initial velocity; attributed to the laser |
| $\sigma_{v_0}$ | 75 m/s | standard deviation of initial velocity |
| $\Delta t$ | 8 ns | time between detector records |

**Figure 3.3: Simplified flight tube and parameters.** Schematic denoting the individual stages and path of a compound with mass-to-charge ratio $m/z$ within a time-of-flight tube between sample plate and detector. Description of parameters with default settings. (Reproduced from Coombes et al., 2005.)

estimation of the variance parameter, one should therefore consider the estimation in units of the detector time rather than directly on the mass.

### 3.2.2 Physics based time-of-flight model

The physics based time-of-flight model, introduced by Coombes et al. (2005), formulates the physical relations of ionization and acceleration towards the detector. Given a particle with specified mass $m$ and charge $z$, the physical time-of-flight model computes the particle's velocity resulting from this acceleration. From the velocity, the instrument parameters, and the settings of the electrical field the particle's path through the flight tube and arrival time at the detector is computed. The settings for the electrical field and other instrument parameters needed to compute the velocity are known to, respectively adjustable by, the operator of the instrument. All parameters in this model are listed in Figure 3.3.

In a flight tube (see Figure 3.3), three phases of flight time have to be discriminated once the particle was hit by the laser. The laser impact will cause the particle to have an initial velocity $v_0$.

The particle will keep this velocity during the *dwell time* $\delta$ which essentially denotes a constant offset, adjustable by the operator. The dwell time mostly corrects for height differences in the tissue across laser shots by giving the plume time to enter the range of the accelerating electrical field (O'Connor et al., 2013). What follows are further phases in the different stages of the flight tube. The initial phases are *focusing time* $t_f$, at which end the particle has the velocity $v_1$, and the *acceleration time* $t_a$, which brings the particle to its final velocity $v_2$. Both changes are achieved by an electrical field with voltages $U_1$ and $U_2$. This velocity is kept during the *drift time* $t_d$, the final and longest phase. Naturally, the total time of an analyte in the flight tube is then the sum of the three time phases $t = t_d + t_a + t_f + \delta$. Except for the dwell time $\delta$, all time steps mainly depend on the mass of the particle allowing the discrimination of masses at the detector.

The velocities are all influenced by the physical dimensions of the instrument and can be computed by exploiting the law of conservation of energy. In detail this results in the following relations. Simple physics states that the work

$$W = Fd = mad$$

equals force $F$ times distance $d$ and force itself equals mass $m$ times acceleration $a$. Moreover, work can be expressed as the change in kinetic energy

$$W_i = \Delta E_i = \tfrac{1}{2}m(v_i^2 - v_{i-1}^2)$$

where $v_i$ and $v_{i-1}$ are the velocities after and before the application of force $F_i$. The electrical force $F_i = zU_i/d_i$ is generated by an electric field of voltage $U_i$ that acts on a particle with constant charge $z$ over distance $d_i$.

These are the ground principles in the physical time-of-flight model and this model allows to compute the physical variables in the different phases. First during the focusing phase one observes a change in kinetic energy

$$\Delta E_1 = \tfrac{1}{2}mv_1^2 - \tfrac{1}{2}mv_0^2 = F_1(d_1 - \delta v_0) \tag{3.3}$$

that is equal to the force generated by the acceleration force $F_1$ which is generated after the particle has travelled a short time $\delta$ at velocity $v_0$ closer to the accelerating grid. Using $F_1 = ma_1$ and solving (3.3) for $v_1$ gives the velocity

$$v_1^2 = v_0^2 + \frac{2F_1}{m}(d_1 - \delta v_0) = v_0^2 + 2a_1(d_1 - \delta v_0)$$

of the particle at the end of the focusing phase. Likewise one obtains from $\Delta E_2$ and $F_2 = ma_2$

$$v_2^2 = v_1^2 + \frac{2F_2}{m}d_2 = v_1^2 + 2a_2 d_2 \tag{3.4}$$

as the velocity after the acceleration. Furthermore, after the acceleration phase the velocity

$$v_2 = \frac{L}{t_d} \tag{3.5}$$

is assumed to be constant and can be expressed as the ratio of distance $L$ and time $t_d$ that is passed. Using (3.4) and (3.5) one obtains

$$t_d = \frac{L}{\sqrt{\frac{2zU_2}{m} + v_1^2}}$$

for the drift time. For the other times in the individual phases the focusing time

$$t_f = \frac{v_1 - v_0}{a_1} = \frac{m}{F_1}(v_1 - v_0)$$

in which an acceleration $a_1$ holds that can be expressed by the force $F_1 = ma_1$. With this velocity known the difference of energies

$$\Delta E_2 = \frac{1}{2}mv_2^2 - \frac{1}{2}mv_1^2 = F_2 d_2$$

in the acceleration phase allows solving for the acceleration time

$$t_a = \frac{md_2}{zU_2}\left(\frac{L}{t_d} - v_1\right)$$

of that phase.

In summary, Coombes et al. (2005) lists the equations

$$
\begin{aligned}
v_1^2 &= v_0^2 + \frac{2zU_1(d_1 - \delta v_0)}{md_1} = v_0^2 + 2a_1(d_1 - \delta v_0) \\
t_d &= \frac{L}{\sqrt{\frac{2zU_2}{m} + v_1^2}} = \frac{L}{\sqrt{2a_2d_2 + v_1^2}} \\
t_a &= \frac{md_2}{zU_2}\left(\frac{L}{t_d} - v_1\right) = \frac{1}{a_2}\left(\sqrt{2a_2d_2 + v_1^2} - v_1\right) \\
t_f &= \frac{md_1}{zU_1}(v_1 - v_0) = \frac{1}{a_1}(v_1 - v_0)
\end{aligned}
$$

for flight times, where $a_1 = \frac{zU_1}{md_1}$ and $a_2 = \frac{zU_2}{md_2}$. Apparently, these values depend on instrument parameters listed in Table 3.3 and the ratio of mass $m$ and charge $z$. The only other variable not known is the initial velocity $v_0$. Coombes et al. (2005) set the particle's initial velocity by simulating it as a normal distributed random variable. Then, a realistic peak shape is obtained given the particle's $m/z$ value and can be used to generate simulated mass spectra. However, equations for flight time are a complex model and the shape is mainly depending on the particle's initial velocity. Therefore, it is not as suited for the case when the peak shape is observed and the $m/z$ value needs to be obtained.

### 3.2.3 Skewed Gaussian function

Alexandrov et al. (2010) relied on a Gaussian function for recovering the $m/z$ value based on the peak shape. However, there are reasons such as the often observed asymmetry of the peak shape to expand this model by introducing parameterized modifications of the Gaussian function (Kempka et al., 2004; Kobarg et al., 2014). While shift and scaling of the function lead to the wavelet transform (Shin et al., 2010; Morris et al., 2008), changing the kurtosis and the skewness of the Gaussian are options to consider (Maass et al., 2011; Morris et al., 2008). This section will mainly focus on the skewed Gaussian function (Azzalini, 1985), as it provides parameters to control skewness. Furthermore, two possible alternatives (Foley, 1987; Lan and Jorgenson, 2001) are reviewed in the next section.

The skewed Gaussian function defined by

$$
f(z;\alpha) = 2\,\phi(z)\,\Phi(\alpha z), \tag{3.6}
$$

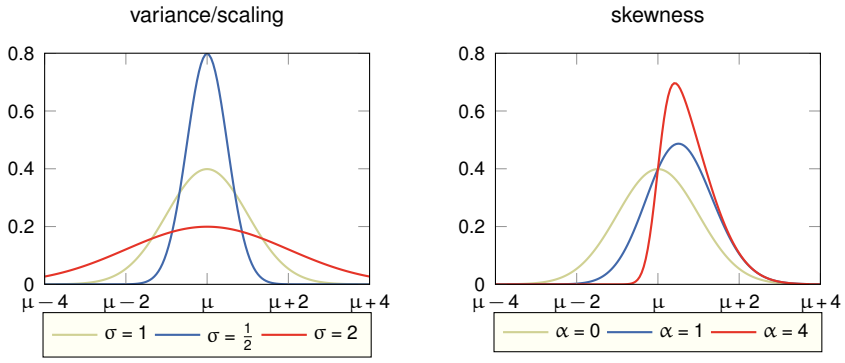



**Figure 3.4: Peak shapes of the skewed Gaussian function.** Variation of scale ($\sigma^2$) and skewness ($\alpha$) is shown for selected values.

is an unimodal probability density function with shape parameter $\alpha$ (Azzalini, 1985). Basically it is the multiplication of the standard normal probability density

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \tag{3.7}$$

and its cumulative distribution

$$\Phi(\alpha z) = \int_{-\infty}^{\alpha z} \phi(\zeta)\, \mathrm{d}\zeta \tag{3.8}$$

integrated up to $\alpha z$. This way the shape parameter $\alpha \in \mathbb{R}$ controls the skewness of the function which vanishes for $\alpha = 0$. Setting $\alpha = 0$ leads to $\Phi(0) = \frac{1}{2}$ and an unskewed Gaussian is obtained. This means, the Gaussian function is a member of this parameterized function family and not a limit (Azzalini, 1985).

Furthermore, the function (3.6) fulfils all criteria of a probability density. Therefore, applying a linear shift $\mu$ and standard deviation $\sigma > 0$ to (3.7) and (3.8)

$$\begin{aligned} f_{\text{skew}}(t;\mu,\sigma,\alpha) &= \frac{2}{\sigma}\,\phi\left(\frac{t-\mu}{\sigma}\right)\Phi\left(\alpha\frac{t-\mu}{\sigma}\right) \\ &= \frac{1}{\pi\sigma}\exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right)\int_{-\infty}^{\alpha\frac{t-\mu}{\sigma}} \exp\left(-\frac{\zeta^2}{2}\right)\mathrm{d}\zeta \end{aligned} \tag{3.9}$$

is obtained. The peak shapes shown in Figure 3.4 demonstrate the influence of the shape parameter $\alpha$ compared to the unskewed Gaussian. The density function (3.9) characterizes a probability distribution for a random variable $Z$ and can be described by its moments $\mathrm{E}(Z^c)$, $c \geq 1$, which

can be computed by the moment generating function (Ross, 2007). For a random variable $Z$ distributed like (3.6) Azzalini (1985) has shown that

$$M_Z(k) = \mathrm{E}(\mathrm{e}^{kZ}) = 2 \cdot \exp\left(\tfrac{1}{2}k^2\right) \Phi(\beta k)$$

with $\beta = \alpha / \sqrt{1+\alpha^2}$ is the corresponding moment generating function. In the shifted and scaled case of (3.9), where $Z = \frac{1}{\sigma}(T-\mu)$, the moment generating function

$$M_T(k) = \mathrm{E}(\mathrm{e}^{kT}) = 2 \cdot \exp\left(\mu k + \tfrac{1}{2}(\sigma k)^2\right) \Phi(\beta \sigma k) \tag{3.10}$$

is obtained by the relation $M_{\sigma Z+\mu}(k) = M_Z(\sigma k) \cdot \exp(\mu k)$. If the $c$-th derivative to $k$ of the moment generating function (3.10) is computed and evaluated at $k = 0$ the moments

$$\mathrm{E}(T^c) = \bar{m}_c = \left. \frac{\partial^i M_T(k)}{\partial k^c} \right|_{k=0}$$

are obtained (Ross, 2007). These expressions can be used to find expectation and variance

$$\begin{aligned} \mathrm{E}(T) &= \bar{m}_1 = \mu + \sigma\beta\sqrt{\tfrac{2}{\pi}} \\ \mathrm{Var}(T) &= m_2 = \sigma^2\left(1 - \tfrac{2}{\pi}\beta^2\right) \end{aligned} \tag{3.11}$$

of the function. The third and fourth central moments $\mathrm{E}(T-\mathrm{E}(T))^3 = m_3$ and $\mathrm{E}(T-\mathrm{E}(T))^4 = m_4$ can also be calculated with the same method and lead to standardized skewness

$$\gamma_T = \frac{\sqrt{2}(4-\pi)\alpha^3}{(\pi + (\pi-2)\alpha^2)^{\frac{3}{2}}} \tag{3.12}$$

and standardized kurtosis

$$\kappa_T = 3 + \frac{8(\pi-3)\alpha^4}{(\pi + (\pi-2)\alpha^2))^2} \tag{3.13}$$

of the function (Azzalini, 1985). Note they depend on the shape parameter $\alpha$ only. The limits of (3.12) and (3.13) as functions of the shape parameter $\alpha$ are

$$\begin{aligned} \lim_{\alpha\to\pm\infty} \gamma_T(\alpha) &= \frac{\sqrt{2}(4-\pi)}{(\pi-2)^{\frac{3}{2}}} \qquad \gamma_T(\alpha=0) = 0 \\ \lim_{\alpha\to\pm\infty} \kappa_T(\alpha) &= \frac{3\pi^2 - 4\pi - 12}{(\pi-2)^2} \quad \kappa_T(\alpha=0) = 3. \end{aligned}$$

Having established the theoretical properties of the parameters, a method to reliable access them given the data needs to be used. With the method of moment estimation, Arnold et al. (1993) solve (3.11) and (3.12) for the unknown parameters and obtain

$$\hat{\mu} = \hat{m}_1 - \sqrt{\frac{2}{\pi}}\left(\frac{\hat{m}_3}{b_1}\right)^{\frac{1}{3}} \quad \hat{\sigma}^2 = \hat{m}_2 + \frac{2}{\pi}\left(\frac{\hat{m}_3}{b_1}\right)^{\frac{2}{3}} \quad \hat{\beta} = \left(\frac{2}{\pi} + \hat{m}_2\left(\frac{\hat{m}_3}{b_1}\right)^{-\frac{2}{3}}\right)^{-\frac{1}{2}} \tag{3.14}$$

where $b_1 = \sqrt{2/\pi}(4/\pi - 1)$ and

$$\hat{m}_1 = \frac{1}{n}\sum_{i=1}^{n} t_i \quad \hat{m}_2 = \frac{1}{n-1}\sum_{i=1}^{n}(t_i - \hat{m}_1)^2 \quad \hat{m}_3 = \frac{n}{(n-1)(n-2)}\sum_{i=1}^{n}(t_i - \hat{m}_1)^3 \tag{3.15}$$

are the bias corrected moment estimators (Klemens, 2009). Within the defined window the estimates can be computed to obtain parameters suiting the observed peak shape. Since overlaps might exist, numerical optimization of the peak shape is done using the estimates (3.14) as initialization. Alternatively, the truncated expectation-maximization (EM) algorithm by Arnold et al. (1993) can be considered.

### 3.2.4 Modified Gaussians and truncated exponentials

In the following, some of the functions used in life science to fit the peak shape will be reviewed. As mentioned in the introduction, these constitute good candidates for related spectrometry methods, but application in MALDI mass spectrometry is unusual (Di Marco and Bombi, 2001). Their ability to model MALDI peak shapes will be compared with performance of the skewed Gaussian function.

#### Exponentially modified Gaussian

In chromatography, the *exponentially modified Gaussian* (EMG) was introduced by Foley (1987) to model a chromatographic peak. The EMG function is quite popular in chromatography as

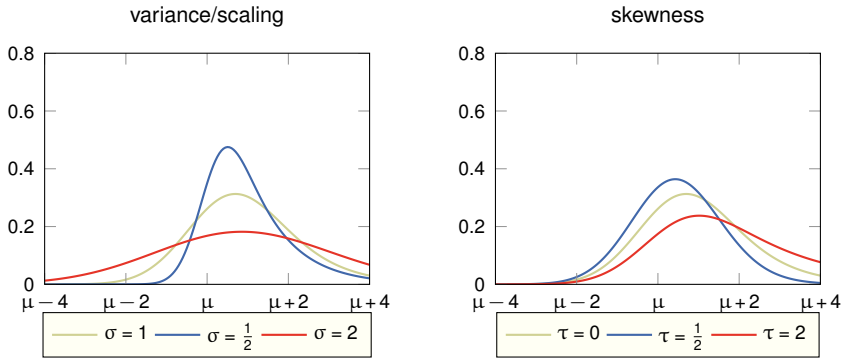



**Figure 3.5: Peak shapes of the exponentially modified Gaussian function (EMG).** Variation of scale ($\sigma^2$) and shape ($\tau$) is shown for selected values.

shown by the list of publications provided by Di Marco and Bombi (2001) It has been used at least once for MALDI-TOF data (Mantini et al., 2008). The function

$$\begin{aligned} f_{\mathrm{EMG}}(t;\mu,\sigma,\tau) &= \frac{1}{\tau\sqrt{2\pi}} \exp\left(\frac{\sigma^2}{2\tau^2} - \frac{t-\mu}{\tau}\right) \int_{-\infty}^{\frac{t-\mu}{\sigma}-\frac{\sigma}{\tau}} \exp\left(-\frac{y^2}{2}\right) \mathrm{d}y \\ &= \frac{1}{2\tau} \exp\left(\frac{\sigma^2}{2\tau^2} - \frac{t-\mu}{\tau}\right) \operatorname{erfc}\left(\frac{1}{\sqrt{2}}\left(\frac{\sigma}{\tau} - \frac{t-\mu}{\sigma}\right)\right) \end{aligned} \tag{3.16}$$

results from the convolution of a Gaussian function

$$f_{\mathrm{gauss}}(t;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(t-\mu)^2}{2\sigma^2}\right)$$

with parameters $\mu$ and $\sigma > 0$, and an exponential distribution function

$$f_{\mathrm{exp}}(t;\tau) = \frac{1}{\tau} \exp\left(-\frac{t}{\tau}\right)$$

with shape parameter $\tau > 0$. See Figure 3.5 for the influence of the parameters to the function. The function $\operatorname{erfc}(t)$ denotes the complementary error function, which relates as

$$\operatorname{erfc}\left(\frac{-t}{\sqrt{2}}\right) = 2\Phi(t) = \frac{2}{\sqrt{2\pi}} \int_{-\infty}^{t} \exp\left(-\frac{y^2}{2}\right) \mathrm{d}y$$

to the standard normal probability function $\Phi$, i.e. an uncentred and unscaled Gaussian.

Parameter estimation can be done analogous to the skewed Gaussian (Lan and Jorgenson, 2001). A random variable $T$ with the distribution function (3.16) has

$$\begin{aligned} \mathrm{E}(T) &= \bar{m}_1 = \mu + \tau & \mathrm{E}(T-\mathrm{E}(T))^3 &= m_3 = 2\tau^3 \\ \mathrm{Var}(T) &= m_2 = \sigma^2 + \tau^2 & \mathrm{E}(T-\mathrm{E}(T))^4 &= m_4 = 3\sigma^4 + 6\sigma^2\tau^2 + 9\tau^4 \end{aligned}$$

as the expectation and central moments. Therefore, the standardized skewness

$$\gamma_T = \frac{m_3}{\sigma_T^3} = \frac{2\tau^3}{(\sigma^2+\tau^2)^{\frac{3}{2}}} = \frac{2\left(\frac{\tau}{\sigma}\right)^3}{\left(1+\left(\frac{\tau}{\sigma}\right)^2\right)^{\frac{3}{2}}}$$

and standardized kurtosis

$$\kappa_T = \frac{m_4}{\sigma_T^4} = \frac{3\sigma^4+6\sigma^2\tau^2+9\tau^4}{(\sigma^2+\tau^2)^2} = \frac{3+6\left(\frac{\tau}{\sigma}\right)^2+9\left(\frac{\tau}{\sigma}\right)^4}{\left(1+\left(\frac{\tau}{\sigma}\right)^2\right)^2}$$

can be expressed in terms of the parameters defining (3.16). As can be seen, these terms are functions parameterized by the ratio $\rho = \frac{\tau}{\sigma}$. Therefore, the skewness and kurtosis of the EMG also depend on the scale parameter $\sigma$, while in (3.12) and (3.13) for the skewed Gaussian function this was not the case. Furthermore, note that for increasing $\tau$ the position of the mode increases and the amplitude decreases. Most importantly, the skewness has a lower bound of 0, because $\tau > 0$ and the denominator is always positive. Therefore, the Gaussian function $f_{\text{gauss}}$ is not part of the function family, but forms its limit as $\tau \to 0$, likewise $f_{\text{exp}}$ is the limit as $\sigma \to 0$ (Lan and Jorgenson, 2001). Employing the method of moment estimation, the sample estimators $\hat{m}_1, \ldots, \hat{m}_3$ as in (3.15) are used to substitute the parameters of (3.16). This way one obtains

$$\hat{\mu} = \hat{m}_1 - \left(\frac{\hat{m}_3}{2}\right)^{\frac{1}{3}} \quad \hat{\sigma}^2 = \hat{m}_2^2 - \left(\frac{\hat{m}_3}{2}\right)^{\frac{2}{3}} \quad \hat{\tau} = \left(\frac{\hat{m}_3}{2}\right)^{\frac{1}{3}} \tag{3.17}$$

as estimators. An important aspect to note here is, the estimation of $\hat{\tau}$ will have the same sign as $\hat{m}_3$ and as such the restriction of $\tau > 0$ can be violated. In numerical experiments the data of peaks indicated several times a skew of $\gamma_T < 0$ requiring $\tau < 0$.

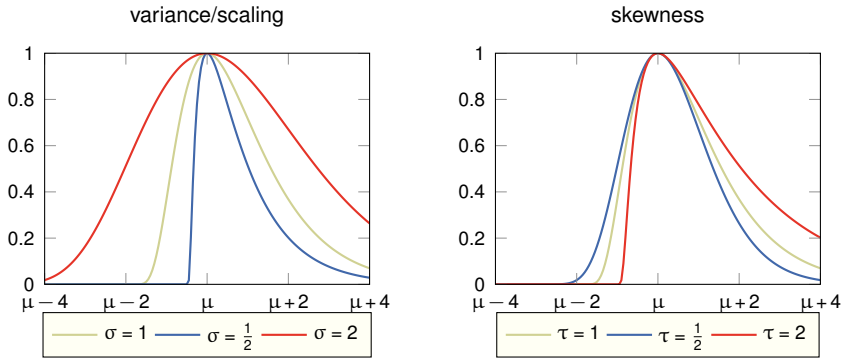



**Figure 3.6: Peak shapes of the Exponential-Gaussian hybrid function (EGH).** Variation of scale ($\sigma^2$) and shape ($\tau$) is shown for selected values; note that the area under the curve of $f_{\mathrm{EGH}}$ is not normalized to unit.

### Hybrid of Gaussian and truncated exponential functions

A hybrid of a *Gaussian and a truncated exponential* function, also called *Exponential-Gaussian hybrid* (EGH), was proposed by Lan and Jorgenson (2001) to account for the asymmetry of peaks. The foundation of this model are two functions

$$f_{\mathrm{gauss}}(t) = h\exp\left(\frac{-(t-\mu)^2}{2\sigma^2}\right)$$

$$f_{\mathrm{trexp}}(t) = \begin{cases} h\exp\left(\frac{-(t-\mu)}{\tau}\right), & \tau(t-\mu) > 0, \\ 0, & \tau(t-\mu) \leq 0, \end{cases}$$

which are combined to

$$f_{\mathrm{EGH}}(t) = \begin{cases} h\exp\left(\frac{-(t-\mu)^2}{2\sigma^2+\tau(t-\mu)}\right), & 2\sigma^2+\tau(t-\mu) > 0, \\ 0, & 2\sigma^2+\tau(t-\mu) \leq 0. \end{cases}$$

Figure 3.6 shows the peak shapes depending on the scale and shape parameters in comparison to the Gaussian profile. The main idea of this function is to simplify the model of chromatography peaks as described by the exponentially modified Gaussian. When a chromatography peak is highly asymmetric, it cannot be fitted by the EMG as discussed by Lan and Jorgenson (2001).

Furthermore, initial parameters for the hybrid function are easier to find. These are the height $h$ of the peak and its position $\mu$. The shape and scale parameters

$$\sigma^2 = \frac{-1}{2\ln\alpha}(h_{+\alpha}h_{-\alpha}) \quad \text{and} \quad \tau = \frac{-1}{2\ln\alpha}(h_{+\alpha} - h_{-\alpha}) \tag{3.18}$$

of the function can be estimated based on the tail values $h_{-\alpha}$ and $h_{+\alpha}$ of the peak shape. In contrast, the description of the function based on statistical moments is not given. So far only numerical approximations exist

$$\begin{aligned} \mathrm{E}(T) &= \bar{m}_1 = \tau\varepsilon_1 + \mu & m_3 &= (3\sigma^2 + 4\sigma|\tau| + 4\tau^2)\tau\varepsilon_3 \\ \mathrm{Var}(T) &= m_2 = (\sigma^2 + \sigma|\tau| + \tau^2)\varepsilon_2 & m_4 &= (3\sigma^4 + 10\sigma^2\tau^2 + 9\tau^4)\varepsilon_4 \end{aligned}$$

where $\varepsilon_c$, $c = 1,\ldots,4$, denotes the numerical integration error (Lan and Jorgenson, 2001). Because the parameters of the function can already be estimated by the shape of the peak, the moments are listed here for the sake of completeness. Lan and Jorgenson (2001) compared the EGH versus the EMG only, considering just chromatography data which is different to MALDI.

### 3.2.5 Function fitting with moment estimation

As described in the previous section, the peak shapes for (3.9) and (3.16) are defined by parameters which can be estimated by the moments. However, for efficiently estimating the peak shape, the traditionally employed moment estimators (3.15) usually reported in literature (Klemens, 2009) are of little use with single mass spectra. The literature estimates are assuming to have a random variable $T$ and observations $t_1,\ldots,t_n$ being realized with a certain probability. In mass spectra it is known how many observations of a certain $m/z$ ratio corresponding to the detector bin $t_i$ were made. Therefore, the number of of each observed ions for an $m/z$ can directly be taken into account. This leads to the introduction of the mean as

$$\tilde{m}_1 = \sum_{i=1}^{n} w_i t_i \tag{3.19}$$

where $w_i$ denotes the frequency with which the mass $t_i$ was observed and $\sum_{i=1}^{n} w_i = 1$. The mean $\hat{m}_1$ is obtained by setting $w_i = \frac{1}{n}$ for all $i = 1, \ldots, n$. Higher decentral moments likewise need to be obtained in a similar manner. For $c = 2, 3$ denote $V_c = \sum_{i=1}^{n} w_i^c$, then

$$\tilde{m}_2 = \frac{1}{1 - V_2} \sum_{i=1}^{n} w_i (t_i - \tilde{m}_1)^2 \quad \text{and} \quad \tilde{m}_3 = \frac{1}{1 - 3V_2 + 2V_3} \sum_{i=1}^{n} w_i (t_i - \tilde{m}_1)^3 \tag{3.20}$$

are the bias corrected moments for weighted observations. Higher estimators from observed data are not needed to compute (3.14) and (3.17).

The peak shape functions will be evaluated based on the spinach data in the next section. For the spinach data multiple detector resolutions are available. This allows to evaluate the effect on the moments and later the fit of the discussed function candidates. Available detector resolutions range from 4 GHz, 2 GHz, …, 62.5 MHz, with the resolution being halved. For economical reasons, the detector resolution is set to 125 MHz in IMS experiments (Shin et al., 2007). This is due to the amount of pixels that are obtained. The fit was computed in two steps. First, the three moments based on the frequency formulas (3.19)–(3.20) were computed for all peaks found in the data. With these, estimates for the shape parameters of the skewed Gaussian function (3.14), and exponentially modified Gaussian (3.17), as well as the estimators for the hybrid function (3.18) are obtained. Second, the resulting shape parameters are used as initial values of a numerical optimization.

The mean values of the corrected moment estimates for the spinach data for all peaks are shown in Figure 3.7. For more robust display of the values, 5 % of the moment estimates were excluded as outliers. As expected, the general trend of the empirical variance in the data is negatively correlated with the decreasing sample rate of the detector. The same is true for the empirical skew. Since the isotope pattern is obscured by decreasing sample rate this in turn results in the increase of an apparent skewness. However, the graph only shows the mean of the absolute value. Since the third moment can take both positive and negative values, this does also occur for the given data. Effectively, this results in a distribution of the skews around zero for all peaks in the data. Therefore, the restriction of $\tau > 0$ for the EMG model (3.16) is violated.

Additionally, the region of the most abundant peak in the spinach spectra data (from $m/z$ 647.530 to $m/z$ 656.068), will be explored in more detail. The fit of the skewed Gaussian can be seen in Figure 3.8A. The initial fits by the moment already match the actual data quite well. However, it is worth noticing, that in the case of 4 GHz, numerical optimization did set the skewness parameter $\alpha$ close to 0 which would lead to a standard normal shape of the peak.

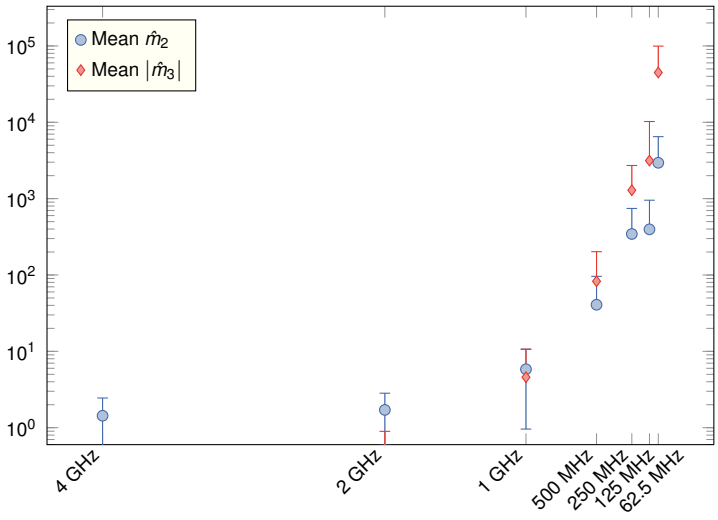



**Figure 3.7: Statistics of moment values for spinach data.** The mean values of all moments estimated for the spinach data at different sample rate shown in a logarithmic scale. Note that the absolute mean of the third moment is shown.

This was not the case for the other resolutions and indicates the data indeed requires this extra parameter for peak fitting.

Parameter estimation for the exponential modified Gaussian is shown in Figure 3.8B. Visually the peak shapes already fit the data well. This suggests the parameters are close to their true values. However, optimization changes the initial parameters, as it can be observed for $\tau$ in the case of 125 MHz. Despite these results, the EMG function has the major disadvantage of the restriction of $\tau > 0$ which cannot be averted by the employed moment estimation. If one has a symmetric peak, it cannot be modelled with any combination of parameters. Symmetric Gaussian profiles can only be achieved with $\tau \to 0$.

The advantage of the hybrid of Gaussian and truncated exponential function is its simplicity in terms of parameterization and estimation. As a result, the overlap of the peak shapes in Figure 3.8C show minimal difference between data and function. Due to the inclusion of height in the parameters of the function, it also does not suffer from the problem of overlapping peak hulls that hampered the skewed Gaussian and exponentially modified Gaussian.

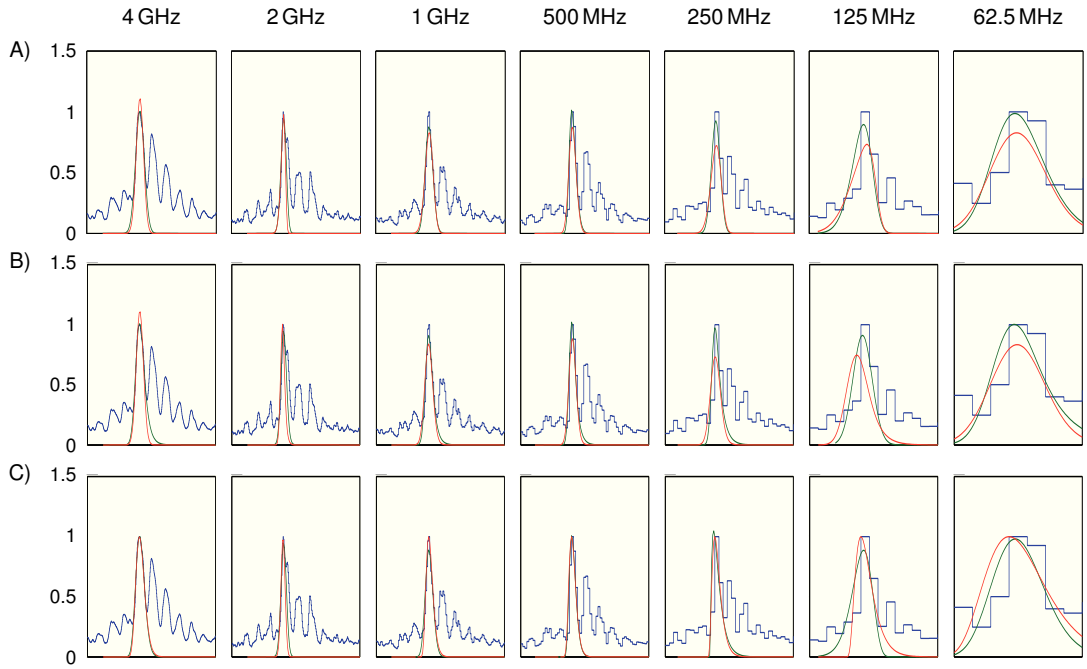



**Figure 3.8: Peak shape function fitting to spinach data.** Mass spectrum is shown in blue, the initial fit by moment estimation in red, the optimized fit in green, all plots depict the data for the window of $m/z$ 647.530 to $m/z$ 656.068. A. Skewed Gaussian distribution. B. Exponential modified Gaussian. C. Hybrid of Gaussian and truncated exponential function.

### 3.2.6 Conclusion of spectrum modelling

Several theoretical options to model a peak are available in literature (Di Marco and Bombi, 2001). Rather than assuming a symmetric Gaussian function, most of the models incorporate parameter to modify the skew. Especially for time-of-flight (TOF) data, the theoretical process of spectrum acquisition can be taken into account. To get a better understanding of the theory that leads to the peak functions, the work by Coombes et al. (2005) was reviewed. From that work one can learn that both the isotope pattern, as well as the conversion from time-of-flight $t$ to mass-to-charge $m/z$ distort the shape of a peak. Alexandrov et al. (2010) assume a Gaussian shape for the peak, which only approximates the underlying theory of a flight tube. Furthermore, related fields of mass spectrometry use more complex models with asymmetric peak shape. Therefore, the skewed Gaussian was introduced to allow more flexibility without disregarding the Gaussian (Kobarg et al., 2014). From theoretical aspect mainly the distortion from isotopes

have an influence on the peak shape. The effect from the conversion from time to $m/z$ is not necessary to model the shapes since detector ticks can be used. Therefore, one expects to have the width of a peak to grow with increasing mass and to have the skew of the peak decrease.

In order to estimate how well the new model performs and to confirm the theoretical expectations an experiment was performed. In this context there is no spatial information included and only single spectra are of interest. As it was explained, a low detector sample rate does not resolve the ions belonging to various isotopes. Consequently, the decreasing sample rate results in an increasing peak width. For the visible peaks at 4 GHz, the peak widths are similar, with decreasing sample rate, the width of the peaks increase. However, the theoretical background leads to the assumption of the skew always being positive. This is even compulsive in the exponentially modified Gaussian model by Foley (1987). With the data at hand this could only be confirmed in part. Since the detector's resolution is not high enough and the spectra are affected by noise, negative skews or at least parameter estimates of them do occur. There exists no motivation for this effect from the theoretical influence of isotopes.

## 3.3 Simulation framework for imaging mass spectrometry data

Individual spectra were the focus in the first part of this chapter. Concluding, a wide range of parameters influences the appearance of the peak shapes. For an IMS dataset, the parameters such as height $a_l$ and position $\mu_l$ of a peak $l = 1, \ldots, p$ in the function (3.1) must be simulated for each spectrum and be based on a class. In Figure 3.9 the workflow of the different steps to obtain such a set of parameters for the model is shown. As input the user has to provide a digital anatomical model in the form of class labels which adhere to a certain hierarchy shown in Figure 3.10. Furthermore, the user can specify a set of mass values that are to be used or have those values be generated accordingly to a probability function with user specified parameters. The anatomical model as well as the $m/z$ values are required as inputs to generate line spectra for each of the pixels belonging to the given classes. The list of input parameters needed to set up the framework is shown in Table 3.1. Following the concept of Coombes et al. (2005), the physical relations within an instrument are modelled and require a set of instrument parameters. This allows to transform the line spectra into spectra with realistic peak shapes using the conversion from flight time $t$ to mass-to-charge ratio $m/z$ as described by Coombes et al. (2005). To generate the baseline function and an amount of background noise in the spectrum, the anatomical model is used again. The concept is shown in Figure 3.11 and will now be explained in more detail. In

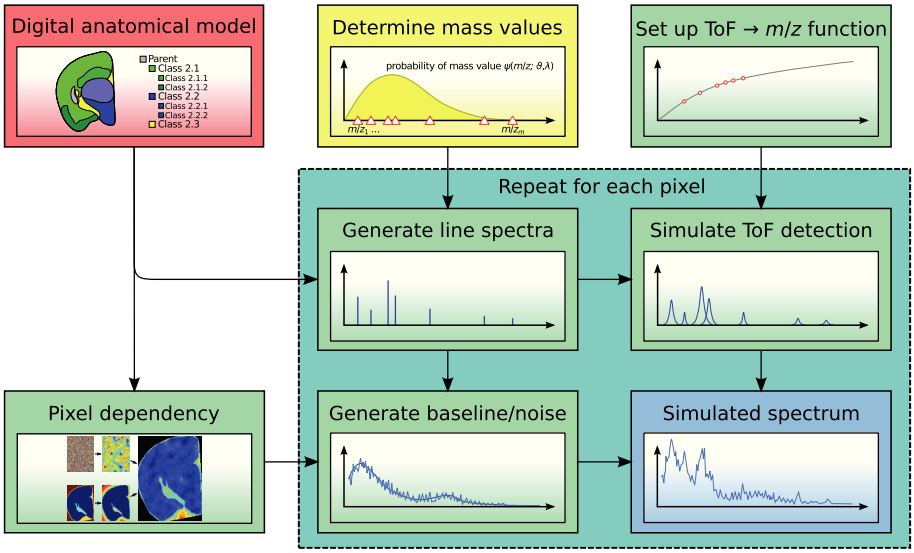



**Figure 3.9: Simulation work flow.** Required user input in red, optional user input in yellow, green self-sufficient methods that work based on user defined parameters or the output of other methods, blue the final output.

brief, the spatially dependent noise and the total ion count of the line spectra are used to set up a smooth baseline function $\beta(t)$. Around the trace of the function, noise is added which depends on the function. The overall outcome is a simulated spectrum for each of the annotated pixels.

### 3.3.1 Allen Brain Annotations

In order to have a realistic anatomical model, the classes are based on the Allen Mouse Brain Atlas (2009). Studies with mouse brain tissue uses this atlas as a reference to compare results with the anatomy (Abdelmoula et al., 2014). The atlas provides detailed annotation for an adult mouse brain in three different resolutions of 200 µm, 100 µm, and 25 µm. These datasets comprise 25,155, 201,284, and 12,803,897 voxels respectively. A spatial resolution of 50 µm is routinely used for analysis of metabolites and proteins (Goodwin et al., 2008). Therefore, the 25 µm resolution data provided by the Allen Institute is used to create a subset with 50 µm resolution by taking only every second pixel in $x$ and $y$ direction in the coronal plane. For the $z$ direction every third pixel is taken into account, giving a resolution of 75 µm, in order to arrive at the goal of one million voxels.

| Name | Description |
|---|---|
| $c = 1, \ldots, C$ | number of hierarchy levels in data |
| $\mu_0^{(c)}, \sigma_\mu^{(c)}$ | mean and standard deviation of nominal mass values per hierarchy level $c$ |
| $p_c = \mu_0^{(c)}/5000$ | number of mass values per hierarchy level $c$ |
| $h_0)$ | base abundance for dataset |
| $\sigma_a$ | maximal deviation of abundance |
| $r$ | convolution radius for pixel dependency |
| $a_w$ | base intensity of pixel dependency |
| $\mu_\beta$ | location of baseline maximum |
| $\lambda_\beta$ | slope parameter for baseline |
| $\sigma$ | variance of additive noise |
| $q$ | number of baseline spline points |

**Table 3.1: Parameters for simulation framework.** Description of parameters with default settings.

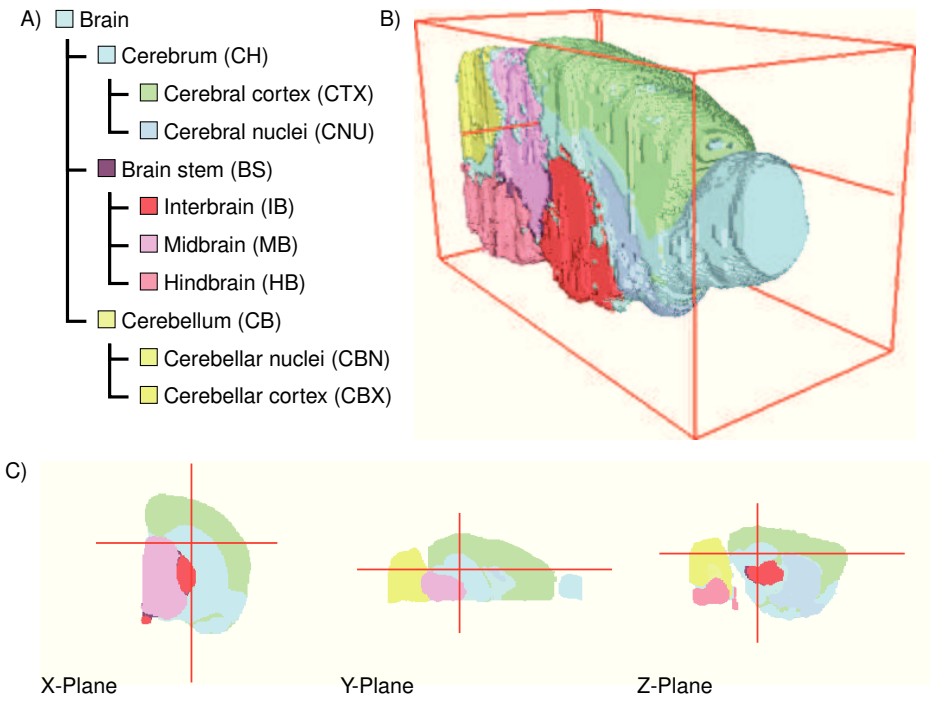



**Figure 3.10: Class annotations and volume rendering.** A. Hierarchy of the annotations based on Allen Mouse Brain Atlas (2009). B. Volume rendering of said labels. C. Virtual sections through an arbitrary point.

From the annotations provided by the Allen Institute the three top anatomical structures are considered. These split off the tissue region along with their two immediate sub-regions. Namely, the regions are *cerebrum* (CH), *brain stem* (BS) and *cerebellum* (CB) which each split into two to three finer sub-regions. The hierarchy is visualized in Figure 3.10 for the first three levels. A class $k$ is therefore associated with the direct parent class $\kappa = \kappa(k)$ and the number of ancestors $c = c(k)$, also called hierarchy level. For example, the *cerebral cortex* (CTX) is $k = 3$, $\kappa = 2$, denoting *cerebrum*, and $c = 3$, as it has two ancestors.

Additionally to the annotated pixels, the measurement area in each coronal section was added. These new pixels in the dataset represent non tissue area measured by the machine. The spectra in this background class do not contain any peaks and only consist of a noisy baseline. The number of classes for the two immediate sub-regions following the top level structure is $K = 12$. Utilizing one more layer of sub-regions, as was done for the simulation, the number of classes $K = 24$. The number of annotated pixels from the atlas is $n_1 = 1{,}066{,}910$, further $n_0 = 134{,}105$ pixels belong to the background class.

### 3.3.2 Simulation of nominal masses

For the whole dataset, a global list $M$ of molecules with nominal masses $\mu_l \in M$, $l = 1, \ldots, p$. was created in the following manner. The nominal masses $\mu_l$ result from a negative binomial distribution with density

$$\psi(\mu_l; \vartheta, \lambda) = \binom{\vartheta + \mu_l - 1}{\mu_l} \lambda^{\vartheta} (1-\lambda)^{\mu_l} \tag{3.21}$$

defined by probability $\lambda \in (0,1)$ and number of trials $\vartheta$. With this parameterization it is not clear how to choose $\lambda$ and $\vartheta$ to receive nominal masses in a certain area of the mass axis. However, the distribution defined by (3.21) can also be characterized by the moments, namely the mean $\mu_0 = \frac{\vartheta(1-\lambda)}{\lambda}$ and the variance $\sigma_\mu^2 = \frac{\vartheta(1-\lambda)}{\lambda^2} = \frac{\mu_0}{\lambda}$. This allows to substitute the parameters

$$\vartheta = \frac{\mu_0^2}{\sigma_\mu^2 - \mu_0} \quad \text{and} \quad \lambda = \frac{\mu_0}{\sigma_\mu^2} \tag{3.22}$$

in (3.21) by choice of suitable $\mu_0$ and $\sigma_\mu$.

Initially it was planned to use individual mass lists $M_k$ for each the different annotated classes $k = 1, \ldots, K$. However, that approach led to too few common mass positions. This in turn pro-

duced classes that are easy to discriminate. Therefore, the mass positions have to be predefined for the whole dataset. To allow variability of mass positions based on the anatomical structures, there will be $C = 4$ mass lists $M_c$ for the chosen hierarchy levels, where each $M_c$, $c = 1,\ldots,C$, has its own parameter pair $\mu_0^{(c)}$, $\sigma_\mu^{(c)}$ defining the the substitute parameters (3.22). This way, each $M_c$ will contain $p_c$ masses generated at random using (3.21). The number of masses per hierarchy level is fixed as $p_c = \mu_0^{(c)}/5000$ which will generate more lower masses than higher ones.

### 3.3.3 Generation of line spectra depending on class and mass

With the mass positions being modelled for the entire dataset, the next step is to determine a line spectrum

$$s^k(t) = \sum_{l=1}^{p} \alpha_{kl} \cdot \delta(t - \mu_l) \tag{3.23}$$

for each class $k = 1,\ldots,K$. This is done via the class hierarchy in the following recursive manner. In general, one can observe in experiments, that the abundance of a peak decreases with the $m/z$ value (Shin, 2006). Therefore, as an initialization with a base abundance of $h_0$ the logarithmic function

$$h_l = h(\mu_l; h_0) = h_0 \log_{\mu_l} 2 \tag{3.24}$$

is used, which is dependent from mass $\mu_l$ for the top level structure *brain*. At this point, the use of a logarithmic function is by no means mandatory and can be replaced with random values or known abundances as well. In so far, (3.24) equals a vector of abundances $(h_1,\ldots,h_p)^T$. To compute the random height of a peak, based on a parent line spectrum, the function

$$\alpha(h, \sigma_a) = \max\{h \cdot (\eta \cdot \sigma_a + 1), 0\}, \tag{3.25}$$

is used where $\eta \sim N(0,1)$ is a standard normal random number and $\sigma_a$ an allowed variance. This enforces a hierarchy by using the value of the parent structure as expectation. From the deterministic initialization (3.24) the line spectrum $s^1(t) = \sum_{l=1}^{p} \alpha_{1l} \cdot \delta(t - \mu_l)$ for *brain* is computed by (3.25) $\alpha_{1l} = \alpha(h_l)$ and $\sigma_a$ being dependent on which of the $C$ mass lists $\mu_l$ belongs to. For all other classes $k = 2,\ldots,K$, (3.25) is a function $\alpha_{kl} = \alpha(\alpha_{\kappa l})$ depending on the parent's value.

However, evaluation of the noise level showed, that the variance parameter $\sigma_a$ should be different for both $k$ and $l$. If this was not the case, the noise in the data was not high enough,

as the peak intensities were too similar. Here, the hierarchy level $c = c_k$ was used to generate individual deviations $\sigma_a^{(kl)}$. Essentially, the deviation parameter $\sigma_a^{(kl)}$ is based on the class' parent deviation parameter $\sigma_a^{(\kappa l)}$. To do this for the other classes, the function

$$\sigma_a^{(kl)} = \begin{cases} \sigma_a^{(\kappa l)} + \varepsilon(\sigma_a^{(\kappa l)}/2^{c_k}), & \text{if } \mu_l \in M_{c_k}, \\ \sigma_a^{(\kappa l)}, & \text{else,} \end{cases}$$

is used, where $\sigma_a^{(\kappa l)}$ is the parent's deviation and $\varepsilon(\lambda)$ is an exponentially distributed random number with density $\psi(\varepsilon;\lambda) = \frac{1}{\lambda}\exp(-\frac{\varepsilon}{\lambda})$. The rationale to use an exponential random number is the heights of peaks within a sub-class should be similar to the heights in parent class. This procedure is repeated for the sub-regions, using the respective abundance values of the parent class of course. The choice of creating all $\sigma_a^{(kl)}$, for $k = 1,\ldots,K$ and $l = 1,\ldots,p$ is quite arbitrary and not modelled with physical observations in mind, but creates datasets with realistic level of noise.

As the function (3.25) allows zero intensities, optionally the value of the deterministic function (3.24) can be used instead. This avoids omitting a peak, as for subsequent classes the intensity would also be zero. However, the removal of a peak from a certain class can also be a feature specific to the class. In the base classes where $c \leq 2$, this should not be used, as good segmentation results can then be achieved simply on the presence of a given peak.

Finally, the individual spectra $i = 1,\ldots,n$ are then constructed based on the $k_i$-th line spectrum (3.23) of the pixel's class in the dataset. This is done by creating $n \cdot p$ standard normal random numbers $\eta_{il}$, one for each mass in the spectrum, shift each by 1 and take the absolute value of it. This creates the individual pixel line spectrum

$$s_i(t) = \sum_{l=1}^{p} |1 + \eta_{il}| \cdot \alpha_{k_i l} \cdot \delta(t - \mu_l)$$

for which the detection with the physical TOF model is simulated, resulting in each peak being distorted.

### 3.3.4 Generation of spatial dependency

One main goal of the simulation is the creation of a realistic imaging dataset where spatial dependency of the spectra is assumed and the influence of a matrix layer is considered. For line spectra, the peak abundance (3.25) only depends on the class. Real-life datasets usually show

strange effects near the edge of a tissue that mostly affect the baselines found in the spectra (Norris et al., 2007; McDonnell et al., 2008). Therefore, special treatment for the pixels near the border of the tissue needs to be simulated. This is achieved by two separate effects that are combined as shown in Figure 3.11. The digital anatomy image $v$ with only pixels from the tissue section is used to recreate one of these effects. All those pixels belonging to either the matrix class or only the top level structure brain were defined as not belonging to the tissue section. The *bwdist* function $\mathcal{D}_d(\cdot)$ from the MATLAB image processing toolbox is then used to compute the distance $d(\cdot,\cdot)$ of each pixel from $v$ to the boundary of the section. A smooth transition from the border into the tissue was achieved using the quasi-Euclidean distance function

$$d(s_1,s_2) = \begin{cases} |x_1-x_2| + (\sqrt{2}-1)|y_1-y_2|, & \text{if } |x_1-x_2| > |y_1-y_2|, \\ (\sqrt{2}-1)|x_1-x_2| + |y_1-y_2|, & \text{else,} \end{cases}$$

where $(x_1,y_1)$ and $(x_2,y_2)$ are the pixel coordinates of two spectra $s_1,s_2$. The function $d(\cdot,\cdot)$ applied to a unit circle resembles an octagon shape.

Further spatial dependency is created by simulation of a matrix layer within each 2D section. The matrix layer is represented as a cloud texture. First, for each pixel at $(x,y)$ a uniformly distributed random number $u_{xy}$ in the interval $[0,2]$ is generated. In the next step, a circular convolution filter $\mathcal{F}_r(\cdot)$ with radius $r$ is applied to the image $u$. This introduces a spatial dependence between the formerly independent random pixels and results in

$$u' = \frac{\pi}{2}\big(\mathcal{F}_r(u) - 1\big) \tag{3.26}$$

as a new image which mimics a cloud texture.

Finally, the spatial dependency value is calculated by merging the distance transformed image

$$v' = \Big(\sqrt{r} + \log\big(1 + \mathcal{D}_d(v)\big)\Big)$$

and convoluted image (3.26) to

$$w_{xy} = a_w(1 - u'_{xy}) \cdot v'_{xy} \tag{3.27}$$

as the combined spatial dependency with a given base intensity $a_w$. Even though the goal is to simulate a three dimensional dataset, there is no spatial interaction into the $z$-direction. This is motivated by the fact that sample preparation requires individual matrix application of two dimensional tissue sections.

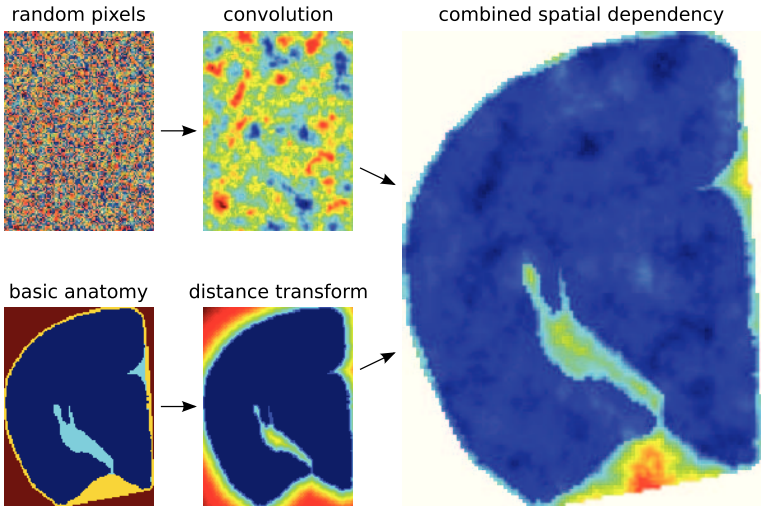



**Figure 3.11: Spatial pixel dependency.** Random values for each pixel is made interdependent by convolution; in parallel the basic anatomy is used to create a bleeding-out effect; in the last step the two inputs are combined to create a spatial dependency for the tissue section.

### 3.3.5 Baseline simulation

In MALDI-TOF spectra, the baseline usually plays a major role, especially those in imaging datasets. Experiments by Schwartz et al. (2003) have shown, that the shape of the baseline is influenced by the used matrix. However, a proper baseline simulation is usually excluded in the available software packages. For real-life data, standard preprocessing packages remove the baseline in an empirical way (Norris et al., 2007; Shin et al., 2010). In this manner, a pointwise vector is removed from the signal. For simulation this approach is not feasible, as it would require a large number of random numbers that are highly correlated.

The use of large number of random numbers can be avoided if the baseline is represented by a smooth function, that depends on a small set of parameters. This smooth function is described by House et al. (2011) with the exponential function $\beta(t;a,\lambda) = a\exp(\lambda t)$, with noise depending on the function value in $t$. However, the use of this particular function as described in the paper seems only to be good to simulate few spectra. In the context of simulating large imaging datasets, the approach does not work well. When the mean spectrum of the simulated dataset is calculated, the shape of the exponential function is too obvious. In real-life datasets this is hardly the case. Furthermore, the function is rather restricted and one cannot model certain

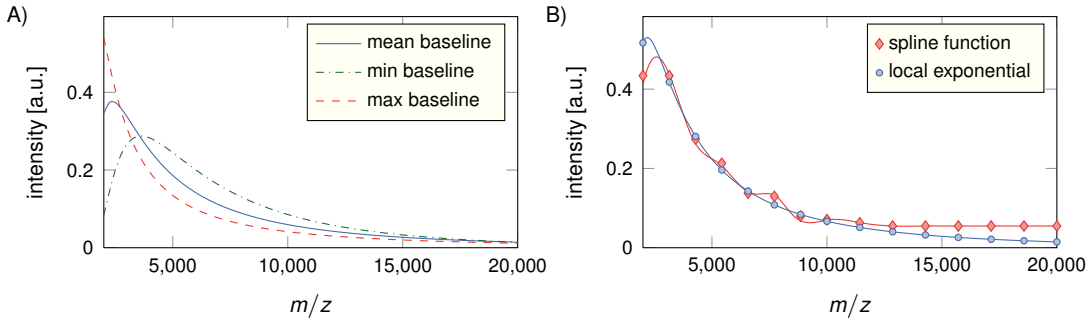



**Figure 3.12: Baseline simulation scheme.** A. Simulation of parameters for Wald distribution results in smooth functions determining the trend of the baseline. B. Interpolation of shifted points from the linear trend function results in a curvy baseline function.

characteristics such as a high, almost polynomial bulge in the low mass range. This can be achieved by increasing the number of parameters of the function. The resulting model

$$\beta(t;a_1,a_2,\lambda_1,\lambda_2) = a_1 \exp(\lambda_1 t) + a_2 \exp(\lambda_2 t) \tag{3.28}$$

is popular in literature (Williams et al., 2005; Sun and Markey, 2011) because with four parameters $a_1, a_2, \lambda_1, \lambda_2$ it is highly flexible to match empirically found baselines. Even though this function family allows more control over the shape of the functions to fit real-life data, simulation of parameters using the ranges obtained by the parameter estimates from real-life data does not generate realistic baselines. Simulation of the parameters usually results in implausible shapes as certain combinations of parameters may not be valid. Furthermore, the choice of $a_1, a_2, \lambda_1, \lambda_2$ is not intuitive beyond utilizing the estimated ranges. Reformulation of the parameters in terms of the derivatives in the origin and of the position of the maximum make the definition more predictable, but still allows too many degrees of freedom and with that invalid parameter combinations.

Since the most intuitive way is to describe the location of the maximum with a parameter $\mu_\beta$, a function allowing this should be used. A function that fulfils this requirement is the probability density of the Wald distribution. The Wald distribution is also known as the inverse Gaussian and its density

$$\beta(t;\mu_\beta,\lambda,a) = \frac{a}{\sqrt{2\pi\lambda t^3}} \exp \frac{-(t-\mu_\beta)^2}{2\lambda\mu_\beta^2 t} \tag{3.29}$$

is defined by only three parameters $a, \mu_\beta, \lambda$, yet it models the trend found in real-life data. The possible trends for the function are shown in Figure 3.12A. The interpretation of (3.29) is then how likely it is to observe ions belonging to the matrix. Control over its maximum is obtained with $\mu_\beta$ and the parameter $\lambda$ which mainly controls the slope. For the amplitude parameter $a$ the pixel value of the spatial dependency image (3.27) is taken. In contrast to the parameters needed for the exponential function (3.28) this allows for specific control over the location and shape of the baseline functions.

### 3.3.6 Spectrum-wise noise level

Most of the standard denoising algorithms assume the noise to be Gaussian distributed (Alexandrov et al., 2010; Shin, 2006). However, when the physics of the detector is taken into account, this assumption does not hold. As the detector counts the number of ions arriving, the error should be Poisson distributed (Keenan and Kotula, 2004; Piehowski et al., 2009). Furthermore, one can observe the noise level within real-life spectra differs and appears to depend on the mass ranges (Shin et al., 2010; Kwon et al., 2008). House et al. (2011) model this effect by using the baseline function as parameter for the random noise. This way, mass dependent noise is simulated by $Z \sim b \cdot (\eta_\sigma + 1)$ where $\eta_\sigma$ is a normal random variable with variance $\sigma$ and $b$ the value of the baseline. As the Poisson distribution converges against the Gaussian for large numbers, this is a valid form for generating Poisson distributed random numbers. This approach produces spectra with a reduced level of noise in the high mass range and an almost chaotic appearance for compounds with $m/z$ being less than 10,000.

Before doing so, another grade of distortion to the general trend of the baseline function adds more realism to the data. This is done by a set of $q$ supporting points $\zeta_1, \ldots, \zeta_q$ for the entire dataset. The points spread evenly through the mass range to be simulated. In each baseline that is to be generated, first a small shift is added to the standard positions in the mass range. Then the heights $\beta_\nu = \beta(\zeta_\nu)$, $\nu = 1, \ldots, q$, are calculated from the baseline trend function (3.29) and to be used as the parameter the probability density

$$\phi(k; \beta_\nu) = \frac{\beta_\nu^k}{k!} \exp(-\beta_\nu)$$

from which a Poisson random number $K$ is generated. Using the supporting points and the random heights $K$, a spline function interpolating them is found. Figure 3.12B demonstrates this concept. The interpolated spline function is then used as baseline for a specific spectrum.

Of course, non-negativity for this function and a decrease in its end tail needs to be ensured. However, this approach leads to the desired high variability of baselines, which is usually found in imaging datasets (Norris et al., 2007).

## 3.4 Evaluation of simulated dataset

The data was simulated with the framework described in Section 3.3 using MATLAB and directly saved into hierarchical data format (HDF). As described in the introduction of this chapter, the goal was the creation of a dataset with at least one million spectra. This was achieved by taking a fraction of annotated pixels from the Allen Mouse Brain Atlas (2009). The decision to directly save into a HDF file allowed to discard the spectrum from MATLAB's memory directly after simulation. Only one spectrum was kept in memory during the entire process. In total the file occupies 68 GB on hard drive. Loading a file this big into the memory to process with MATLAB is not possible. Even though HDF allows partial reading of the data even in MATLAB, the entire dataset is preprocessed in SCiLS Lab (SCiLS GmbH, Bremen). Once the peak picking was performed in SCiLS Lab, the necessary $m/z$ cubes were identified and the reduced line spectra were analysed in MATLAB.

In Figure 3.13 the outcome of the simulation is displayed. Two simulated spectra of different annotated regions are shown in Figure 3.13A–B. As can be seen in Figure 3.13C, the mean spectra from these regions also differ. Looking at different $m/z$ images for a two-dimensional section, we see that the spatial distribution at the given values differs as well. As reference, a spectrum from the rat brain data is shown in Figure 3.13E, along with the mean spectrum of the described dataset (Figure 3.13F). One can see that the spatial distribution of the real-life data is also similar to the simulation result as seen in Figure 3.13G. Therefore, the principal goal of the statistical simulation is achieved, the appearance of the simulated data matches that of real-life data. In the following, the dataset will be analysed in closer detail. In this dataset the level of noise can be considered as very strong.

### 3.4.1 Spectra preprocessing and peak picking

Standard preprocessing routines as described by Trede et al. (2012b) were carried out with SCiLS Lab, as described in Chapter 2. Each spectrum was individually normalized to the ion count and its baseline was removed. In the original data, the parameter settings generated 216 peaks, of which 194 were in the mass range of 2 to 17 kDa which is typically employed to

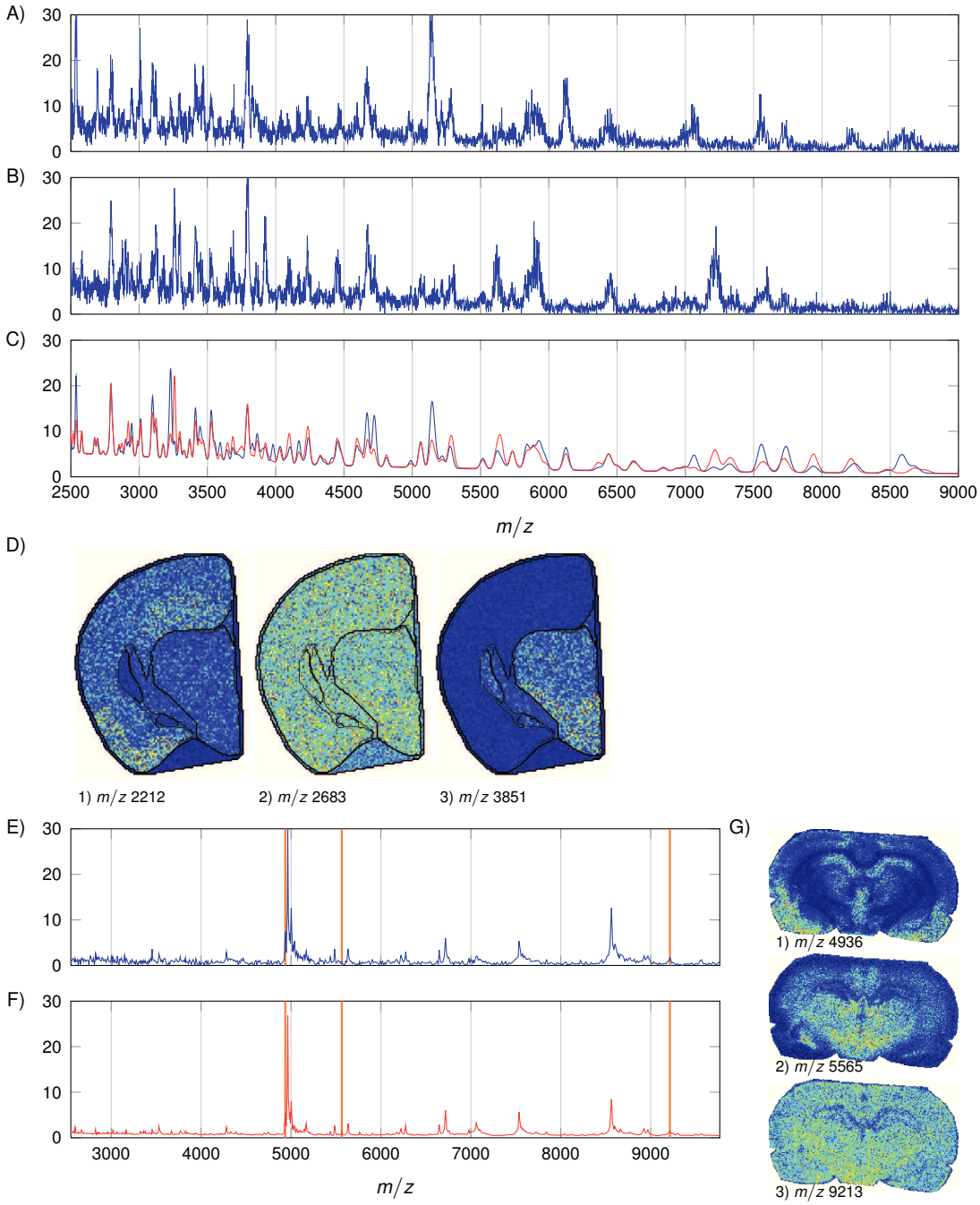



**Figure 3.13: Result of simulation process.** A–B. Two simulated spectra from two different annotations. C. Two mean spectra from those regions. $D_1$–$D_3$. The $m/z$ images of simulated spectra. E-F. Reference spectrum and mean spectrum from real-life dataset. $G_1$–$G_3$. Three $m/z$ images from real-life dataset.

measure proteins and peptides in MALDI-TOF (Goodwin et al., 2008; Rauser et al., 2010a). The peak picker by Alexandrov and Kobarg (2011) was used to obtain peaks for this study. It identified 138 masses as containing peaks, of these, 39 were identical with the true peaks. The remaining 99 peak positions were either near true peaks or marked only noise. Despite this, the segmentation results described in the following subsection show the underlying anatomical structure. This is mainly a result of the employed hierarchy, were several peaks are only necessary to distinguish sub-classes, but their mean intensities are identical to the corresponding super class.

### 3.4.2 Segmentation analysis

The reduced data was read into MATLAB again where three types of segmentation analysis were performed. First, the spatial smoothing was applied as proposed by Alexandrov et al. (2010), second the efficient improvement by Alexandrov and Kobarg (2011), and third plain segmentation without spatial smoothing for comparison. As it can be seen by the comparison of the results shown in Figures 3.14 and 3.15 one obtains drastically different result. If spatial smoothing is applied, the segmentation maps correlate almost perfectly with the anatomical structure. Even though the images for not spatially smoothed data suggest a likewise good correlation with the anatomy, the number of false positives is higher. A large set of pixels is not associated with pixels belonging to the true class and as such increases the number of false negatives. Such holes that are generated by these pixels cannot be visualized in 3D.

For better evaluation, segmentation was also carried out on a single coronal section. The segmentation maps that were generated for the section are shown in Figure 3.16. These results show the performance of the different segmentation approaches. For the agreement scores, the balanced accuracy as described in Section 2.5.4 was used. The segmentation obtained with the method described by Alexandrov et al. (2010) only achieves an agreement of 97.45 % between the ground truth, while the efficient modification proposed by Alexandrov and Kobarg (2011) obtains a near perfect segmentation of 99.62 %. Both methods outperform the segmentation of unsmoothed data, where only 76 % of agreement is obtained.

### 3.4.3 Runtimes / memory for segmentation analysis

In Table 3.2 processing runtimes for the efficient improvement by Alexandrov and Kobarg (2011) are listed, these will be the focus of Section 5.3. As it can be seen, almost three hours

**Figure 3.14: Segmentation maps simulated dataset after spatial smoothing.** The maps show eight classes that are obtained applying segmentation after spatial smoothing.

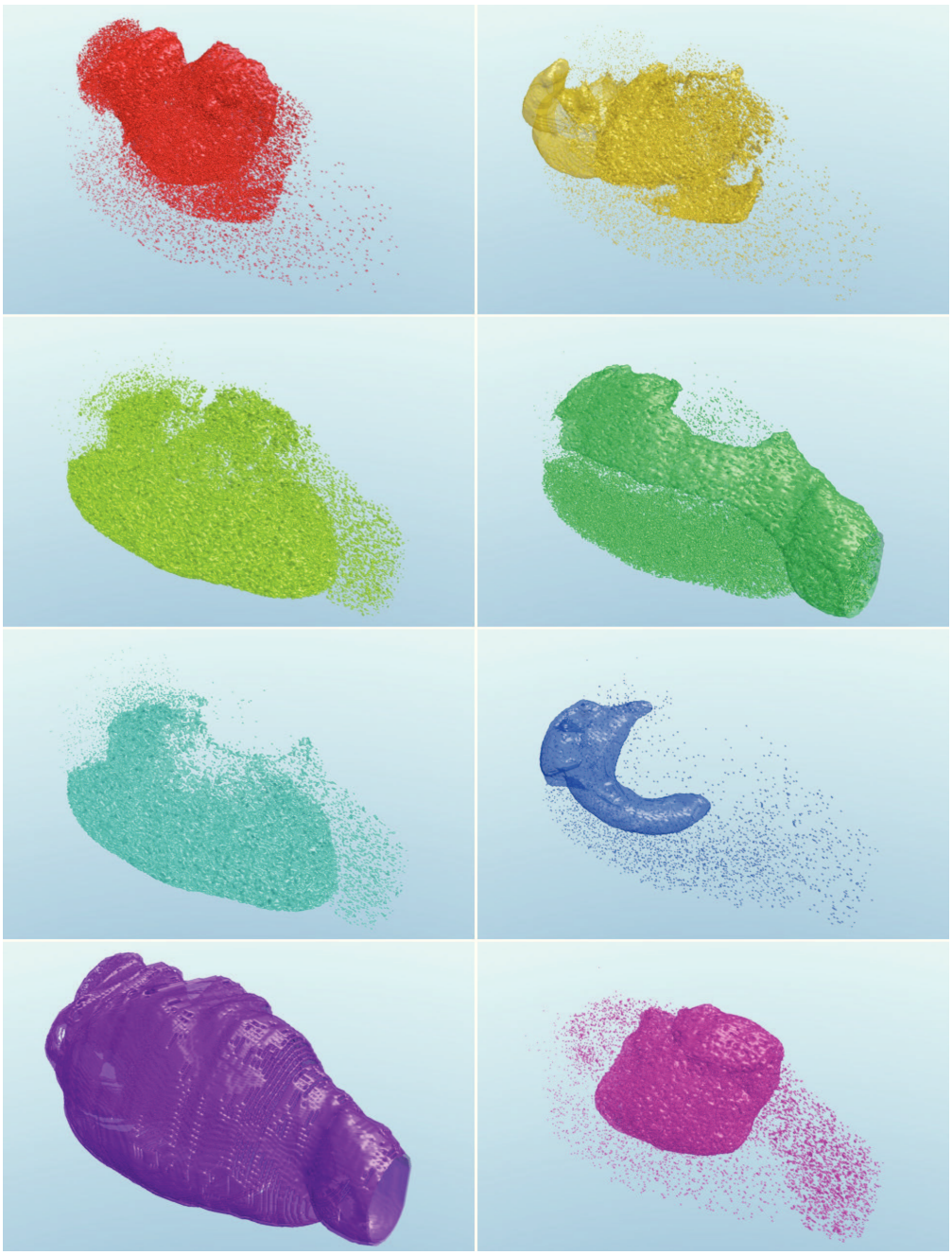


**Figure 3.15: Segmentation maps simulated dataset without spatial smoothing.** Instead of applying spatial smoothing, only preprocessing was applied to the data.

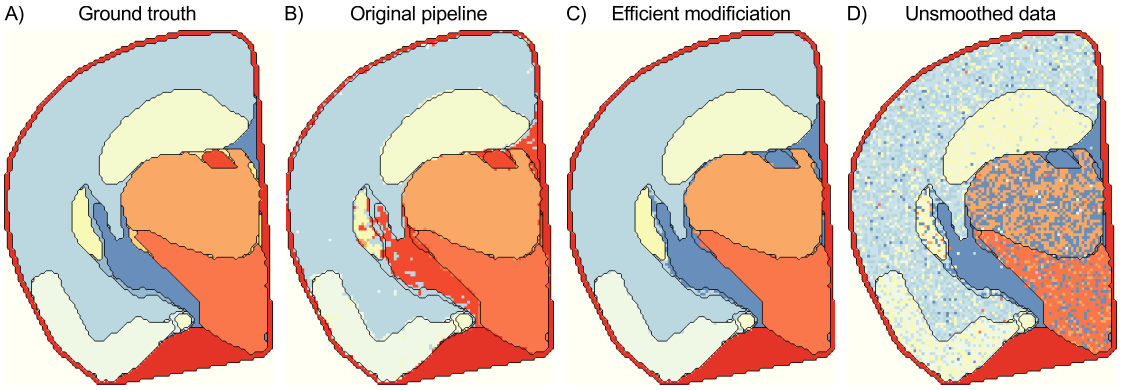



**Figure 3.16: Ground truth and computed segmentation maps.** A. Ground truth labels B. Pipeline proposed by Alexandrov et al. (2010). C. Efficient modification from Alexandrov and Kobarg (2011). D. Segmentation of unsmoothed data. Agreement between (A) and (B): 97.45 % (standard parameters, manually expanded), agreement between (A) and (C): 99.62 % (window size $w = 3$, number of classes $K = 8$), agreement between (A) and (D): 76.00 %.

| Step | Window width | | Description |
|---|---|---|---|
| | $w = 3$ | $w = 5$ | |
| readHdf5 | 10,662.67 s | 10,662.67 s | reading the data from disk |
| medianEnlarge | 28.55 s | 150.87 s | performing boundary correction |
| quantileScaling | 8.26 s | 8.35 s | removing outliers from the data |
| weights | 188.23 s | 395.25 s | computing weights for smoothing |
| FastMap | 2940.74 s | 14,045.85 s | mapping to low-dimensional space |
| $K$-means | 250.00 s | 47.51 s | spatial segmentation |

**Table 3.2: Runtimes of segmentation analysis simulated data.** Allen Brain dataset with $1.2 \times 10^6$ spectra containing very strong noise processed in MATLAB on Intel Core i5 CPU 650 3.2 GHz with four cores and 16 GB Ram.

are necessary to load the data to MATLAB. In contrast to plain clustering with $K$-means of the data, additional computation time for the weights and for the mapping to a lower dimensional space is needed. The required steps to perform preparation of the data to apply the projection into a low-dimensional space for efficient computation, sum up to less than 10 minutes. The computation time for the projection is costly and takes between 49 minutes for a small window of size $w = 3$ and almost four hours for the next bigger window $w = 5$. However, with the data reduced to a small subset including the same information, $K$-means is then faster in finding a solution, probably because the smoothed data points allow the algorithm to converge earlier.

## 3.5 Discussion and conclusion

In this chapter a framework was described which allows the creation of simulated IMS datasets for given annotated pixels. For extended realism, the annotations can be provided in a certain hierarchy. The simulation is flexible in a way that the user can also specify a set of masses or leave this step to the framework as well. If the mass list is not specified, the user only has to determine an approximate average mass and a desired variability. Using the time-of-flight model by Coombes et al. (2005) both the process of obtaining disturbed data by different flight times as well as the calibration of flight time to $m/z$ value is obtained. The approach by Coombes et al. (2005) was introduced to generate single spectra, but not an imaging dataset. Since in imaging datasets the effect of baseline plays a major role, the approach of House et al. (2011) was extended to merge the baseline with the level of noise found in the data. Since their work did not consider IMS either, a spatial dependency of the baseline to the tissue was described in this thesis. The spatial dependency was modelled in that manner, as it is observable for 2D tissue sections. This resulted in a set of statistically simulated spectra for 2D IMS. Furthermore, the simulation framework was used to generate a huge 3D dataset which otherwise takes a high amount of laboratory time to obtain.

The short survey of the simulation quality shows a high level of realism in comparison to data of similar nature. As it is the case with real-life datasets, spatial smoothing is required before segmentation. If this step is not performed, the obtained automatic segmentation corresponds only principally with the given annotation. This becomes more evident by the look at the 2D sections of the analysis. Since the details of reducing peak redundancy and spatial smoothing are the topics to be explored in Chapters 4 and 5, these have been mainly left out in the survey of the data quality.

With the ability to generate huge amount of data that is realistic enough, new theoretical methods can finally be explored. Since the framework allows to set the level of noise, a great flexibility is available to the user. However, the major advantage over real-life data is the known annotation that can be used to test a variety of research goals of which only a few have been presented in this chapter.

# 4 Dimensionality reduction methods

## 4.1 Motivation for this chapter

The interpretation of high dimensional data is often difficult. The influential variables that have a meaning are hidden by the amount of variables measured for one observation (Hastie et al., 2009). Further difficulty arises by the lack of proper visualization, making the data harder to conceive and interpret. Despite the presence of multiple variables, one could decide to rely on univariate analysis only. However, in complex data such as imaging mass spectrometry (IMS), one can often not rely on univariate analysis alone. Multivariate analysis is necessary to detect hidden interactions. Often the number of variables $p$ is higher than needed and one faces the problem to find only the influential $r$ variables. In recent years, multiple dimensionality reduction methods to remove redundancies have been developed (Hastie et al., 2009). Each of these has its own type of application.

The dimensionality reduction methods are slowly gaining popularity in IMS (Jones et al., 2011). In Section 2.4 orthogonal matching pursuit (OMP) was used to reduce the dataset from mass spectrometry to a manageable number of important peaks (Alexandrov et al., 2010). The methods in the first half of this chapter go beyond the simple peak picking. The idea is to describe each spectrum as a weighted sum of a few mixture signatures. This is inspired by the spectrometric structure of metabolites: Each metabolite (peptide, protein, etc.) is reflected by several isotope patterns, that means each metabolite creates a typical spectrum (mixture signature) rather than a single peak. The task is to determine those basic mixture signatures directly from the data. In mathematical terms this asks to determine basis functions in the $m/z$ variable, such that the set of spectra from different measurement locations can be approximated efficiently.

Probably the first choice of dimensionality reduction method in mass spectrometry is the *principal component analysis* (PCA), see for example Coombes et al. (2003); McCombie et al. (2005) or Klerk et al. (2007). With PCA it is easy to identify those variables in the data that have

major contributions to the variability of the data. PCA factorizes the data $X = VZ$ into scores and loadings. Beside visualization of IMS data, PCA can also be used to rank the importance of the peaks and to identify those groups of peaks in the data that are correlated or anti-correlated. The principal components are obtained by singular value decomposition (SVD) which gives an exact decomposition of the data. Often, this is not needed and for PCA the selection of the $r$ most influential components is performed.

In that spirit, one can directly try to obtain an approximation by using a general matrix factorization approach. With a matrix factorization an approximation of the full data matrix $X \in \mathbb{R}^{n\times m}$ is calculated such that $X \approx AS$. For $A \in \mathbb{R}^{n\times r}$ and $S \in \mathbb{R}^{r\times m}$ one aims to find $r \ll \min\{n,m\}$. The two new matrices are most often found with a minimization algorithm (Lee and Seung, 1999). During the search for a solution to solve the minimization problem $\min\|X - AS\|_F^2$ further restriction is enforced to avoid trivial solutions and adapt physical constraints. One such restriction is for example to allow only positive entries in the matrices. This case is called *non-negative matrix factorization* (NMF). Therefore, it is very suitable as a model in IMS as the decomposed model resembles the data more than in the case of PCA. Furthermore, the minimization algorithm allows a flexible incorporation of spatial information (Kobarg et al., 2014).

Both PCA and NMF only focused on finding a good representation of the data based on a data specific low rank decomposition. However, in special cases such as further processing, this is not needed. If one is interested in the similarity of the observations, the method should try to preserve these relations instead. The family of distance preserving dimensionality reduction methods allows this approach. In this thesis the focus will be on FastMap (Faloutsos and Lin, 1995) which offers two applications: First, the data can be replaced by a dictionary independent representation which requires even much fewer entries to store; Second, despite the compressed storage, the new data objects are placed in a space with similarities between them preserved. Therefore, a scatter plot of the new data allows the visual inspection of the similarities. However, the major advantage of the algorithm is that it has linear complexity with respect to the number of spectra.

## 4.2 Principal component analysis

As a tool for multivariate data analysis, the *principal component analysis* (PCA) is one of the most popular ones (McCombie et al., 2005; Klerk et al., 2007; Trim et al., 2008; Sugiura and Setou, 2010). The popularity results from the fact that it is very easy to extract from all $i = 1,\ldots,n$

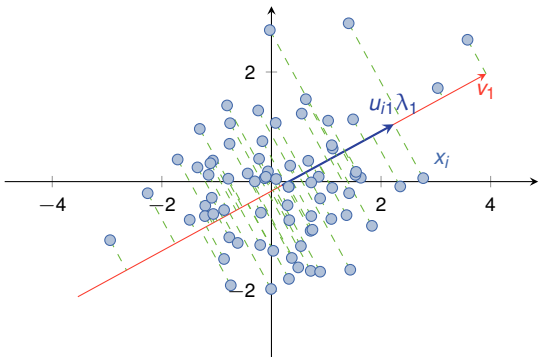



**Figure 4.1: Principal component data projection.** Data $x_i$, $i = 1,\ldots,n$, from a multivariate normal distribution projected onto the first principal component $v_1$. (Reproduced from Hastie et al., 2009.)

elements $x_i \in \mathbb{R}^p$ in a dataset $X$ meaningful vectors $z_i \in \mathbb{R}^r$ which are called *covariates* and explain the variances in the data. This section will summarize the concept and based on Hastie et al. (2009).

The prior introduced spectrum model (2.2) assumes a spectrum $x_i$ to be a linear combination of peaks with observation specific abundance $a$. Therefore, the model can be rewritten as $X = AF$. In practice, it is sufficient to perform PCA on the peak picked data with $p \leq m$ remaining peaks. The goal is then to find a set of $r$-dimensional vectors that encode the redundant information. If the full data needs to be decomposed, simply set $p = m$.

### 4.2.1 Finding principal components

Using the linear model, one assumes the data has an underlying $r$-dimensional structure of the linear form

$$x_i = z_0 + V_r z_i$$

where $z_0 \in \mathbb{R}^p$ denotes the shift and $V_r$ the orthogonal $p \times r$ matrix that transforms the covariates $z_i \in \mathbb{R}^r$. A least squares fit of this model will minimize the functional

$$\min_{z_0, z_i, V_r} \sum_{i=1}^{n} \|x_i - z_0 - V_r z_i\|^2.$$

In the following it is further assumed that the data is centralized and standardized, meaning that

$$\sum_{i=1}^{n} x_i = 0 \quad \text{and} \quad \sum_{i=1}^{n} x_i^2 = 1$$

for each spectrum (Hastie et al., 2009). Then, the minimization problem is connected to the SVD

$$X = U\Lambda V^T \tag{4.1}$$

where $X$ is the data matrix (2.1) with vectors of data in rows, the matrix $U$ is an orthogonal $n \times p$ matrix, $\Lambda$ is a diagonal matrix with ordered diagonal elements $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$ and $V$ is another orthogonal square $p \times p$ matrix. It can be shown that the $\lambda_l$ are the square roots of the eigenvalues of the covariance matrix (Hastie et al., 2009). To obtain the covariance of standardized data one simply has to calculate

$$X^T X = V\Lambda^2 V^T \tag{4.2}$$

due to the previous equation. Therefore, $z_l = Xv_l$ is defined as the $l$-th *principal component*, sometimes also called scores calculated from the $l$-th row of $V$. The variances of these $z_l$ are decreasing with the index $l$. The highest variance is therefore coded in the first component and the least meaningful information in the last component. This allows to pick the components that explain the major variability in the data. Furthermore, the new loadings of the matrix $V$ are uncorrelated, which is often a desired feature. In Figure 4.1 two dimensional data is shown with each dot marking an observation. For this data a PCA has been performed. The red line represents the first loading vector $v_1$ which points into the direction of the highest variance of data. Each data point $x_i$ can be represented by its score $z_i = u_{i1}\lambda_1$, which is the orthogonal projection on the new axis.

### 4.2.2 Principal components for data exploration

There are multiple ways to visualize the new representation of the data. Primarily, it is most common to use a scatter plot of selected components only, called *scores plot*. Each axis uses one of the principal components and the corresponding coefficients are used as coordinates in a scatter plot.

This representation is even more clear, when the input data is from a higher dimensional space. For high dimensional data, the projection of the data into the plane spanned by the

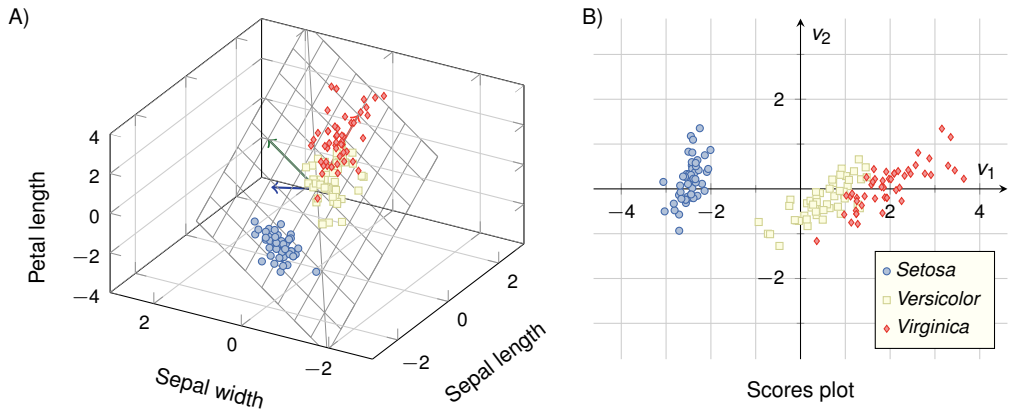



**Figure 4.2: Principal component analysis of Fisher's iris data.** A. First three variables plotted together with the hyperplane spanned by the first two principal components $v_1$, $v_2$. B. The scores plot as the orthogonal projection of the data onto the plane spanned by $v_1$ and $v_2$.

vectors $v_1, \ldots, v_r$ is plotted. Figure 4.2 shows the first three variables of the Fisher Iris data (Fisher, 1936) and the hyperplane spanned by the loading vectors $v_1$ and $v_2$ in originally four-dimensional space. In the scores plot, the separation of the classes can easily be accessed.

Further analysis of the values in $V$ results in better understanding of the influence of the originally measured length on the flower's species. Such an interpretation is visualized with the loadings plot which displays major influences of the features. In the case of the iris data, each loadings plot would be displayed as four items in the plane.

While for most data, the scores plot as shown in Figure 4.2 is the most useful way to display the data, this is not the case for IMS. Since the data has a spatial meaning, the score values are used to create pseudo-images. In Figure 4.3 both the loadings vectors and the score images are displayed for the most influential components.

A special interpretation allow the values of $\Lambda$ in (4.1) or (4.2) respectively (Cichocki et al., 2009). As the entries in the diagonal matrix are sorted by construction, they can be plotted as a decreasing function, where in most cases the plot will reveal drop from high values to lower ones. Furthermore, the $l$-th variance can be divided by the sum of all variances. By this, one can derive the number of needed components to explain a certain amount of variability present in the data. For example one can specify to preserve 95 percent of the variability and disregard all other components. With this approach those components that have little contribution are easily removed from further processing. Such a plot is shown in Figure 4.4 for the rat brain dataset.

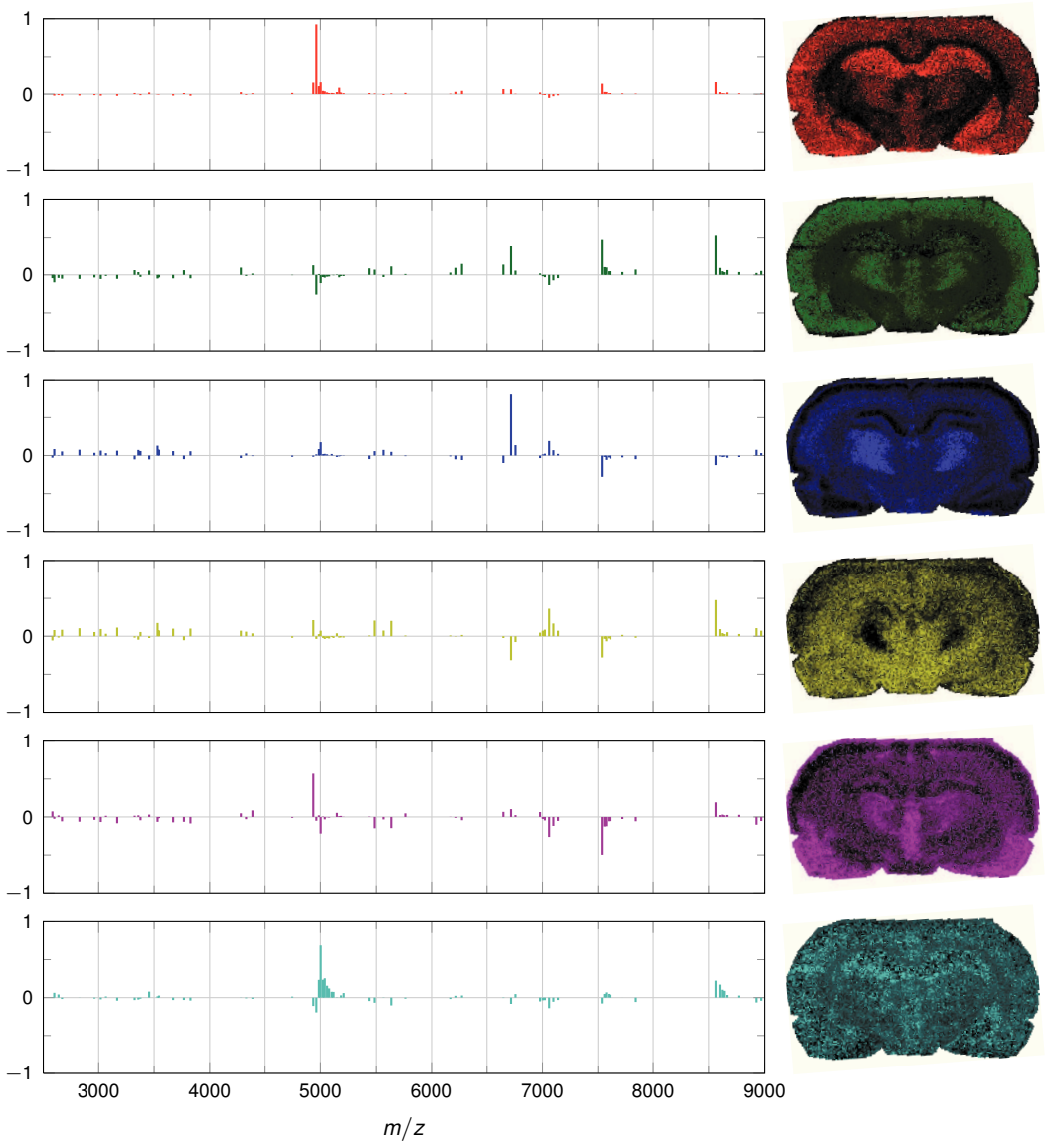



**Figure 4.3: Principal component analysis applied to rat brain dataset.** First six loadings vectors $v_q$ and score images $z_q$ obtained by principal component analysis for $q = 1, \ldots, 6$ in the peak picked rat brain dataset ($p = 71$).

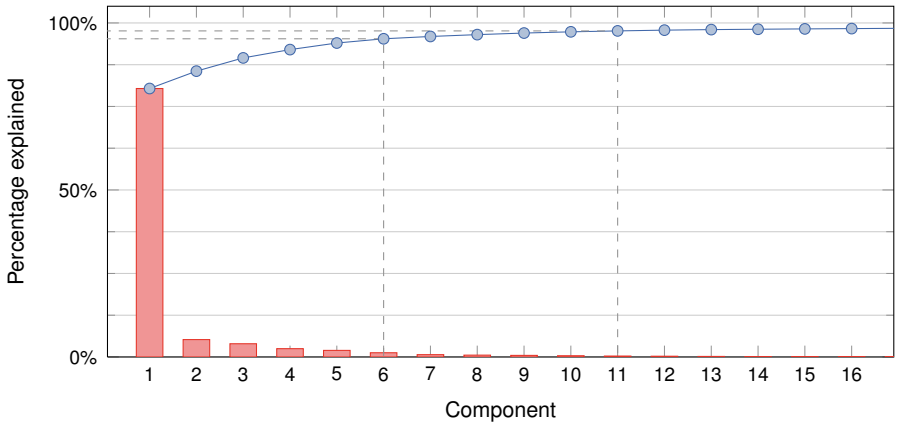



**Figure 4.4: Latent variable number estimation.** The proportion rate and the cumulative proportion rate for the rat brain data. Dashed lines at 95 % and 97.5 % show number of variables needed to explain proportion of variability in data respectively.

### 4.2.3 Conclusion

The concept of PCA has been demonstrated in this section. It is an established method in mass spectrometry (Kemsley, 1996; Coombes et al., 2003) and also used for IMS analysis (McCombie et al., 2005; Klerk et al., 2007; Trim et al., 2008; Sugiura and Setou, 2010). The computation is usually practical on peak picked data only (Jones et al., 2011), as it was done in this section. If no peak picking is performed PCA of the full dataset requires higher memory and takes larger run time. Furthermore, the result is often more difficult to interpret as a high amount of noise present in the unreduced data confounds the interpretability.

Fast computation methods are available to obtain PCA in the context of IMS when the number of spectra is high Klerk et al. (2007). For example, Palmer et al. (2013) have shown the computation of PCA can be accelerated by using randomized projections. Furthermore, Race et al. (2013) reduce the computational cost by a linear acquisition technique and avoids the need to keep the entire data in memory at all time.

## 4.3 Non-negative matrix factorization

In the previous section PCA was introduced as a way to reduce the number of features in a dataset. This was achieved by representing the data as a product of the scores and loadings matrices. The way to find these two matrices was motivated by obtaining an orthogonal coor-

dinate system $V_r$ that maximizes the variances in the individual directions. The loading vectors $v_1, \ldots, v_r$ are dissimilar to the original input. When the data has a predefined structure, such as being grey scale images or ion counts in a mass spectrum PCA helps to identify major contributions. Due to negative intensities, the result is difficult to interpret in comparison to the original data. One way to circumvent this change is to employ *non-negative matrix factorization* (NMF) which also decomposes the data into two matrices of lower rank (Lee and Seung, 1999).

In contrast to PCA, the matrices are found by a minimization algorithm for an objective function rather than by singular value decomposition. This gives a chance to apply further restrictions, such as the non-negativity of the matrices $A$ and $S$. However, the restrictions are not only limited to the matrices being positive, but can also enforce sparsity or spatial smoothness (Hoyer, 2004). A wide range of penalizing expressions is known and to achieve the desired effect suiting the data, one needs to be select the corresponding functional (Sra and Dhillon, 2006).

Matrix factorization is commonly applied in hyperspectral imaging and was also used for IMS by Jones et al. (2011). Further numerical results have been published by Kobarg et al. (2014) and will be presented in this section.

### 4.3.1 Non-negative matrix factorization algorithm

Following the terminology laid out in the previous section, one tries to approximate the data matrix as

$$X \approx AS = \left[a^1 \cdots a^r\right] \cdot \begin{bmatrix} s_1^T \\ \vdots \\ s_r^T \end{bmatrix}$$

where $A$ is a $n \times r$ and $S$ is $r \times m$. However, in contrast to PCA, the solution will be restricted to $x_{ij}, a_{iq}, s_{qj} \geq 0$. In short this will be denoted for the full matrix as $X, A, S \geq 0$. $A$ is called the *mixing matrix* and $S$ is called the *signature matrix*.

Both matrices have a similar interpretation in relation to PCA: The rows $s_q \in \mathbb{R}^m$, $q = 1, \ldots, r$, of $S$ are characteristic spectra; they are also called mixture *signatures* or *sources* (Golbabaee et al., 2010; Hoyer, 2004) and relate to the loadings of PCA. Note that $r \ll \min\{n, m\}$, i.e. the aim is to represent the data matrix $X$ well with a few characteristic spectra. The matrix $A$ contains $r$ columns of size $n$, where $n$ is the number of measurement points. A column $a^q \in \mathbb{R}^n$ of $A$ can therefore be visualized as an image where each pixel corresponds to a measurement point

$i = 1, \ldots, n$. The second major difference to PCA is the restriction of the mixing for each pixel. In practice, one desires to allow $a^q \in [0,1]^n$, such that for all $i = 1, \ldots, n$ it holds $\sum_{q=1}^{r} a_i^q = 1$.

There are different ways of computing such matrix factorizations $X \approx AS$. One can formulate this as the solution to the problem

$$\min_{A,S \geq 0} \frac{1}{2} \|X - AS\|_F^2 \quad \text{subject to } \sum_{q=1}^{r} a_i^q = 1 \text{ for all } i = 1, \ldots, n, \tag{4.3}$$

as proposed by Lee and Seung (2001). The functional (4.3) is of course minimal when the gradient is zero. The derivatives of the norm in (4.3) with respect to $A$ and $S$ can be computed alternating as $\nabla_A = (X - AS)S^T$ and $\nabla_S = A^T(X - AS)$. With the KKT conditions one directly sees the normals

$$0 \stackrel{!}{=} XS^T - ASS^T \Leftrightarrow \frac{XS^T}{ASS^T} = 1 \quad \text{and} \quad 0 \stackrel{!}{=} A^T X - A^T AS \Leftrightarrow \frac{A^T X}{A^T AS} = 1$$

that direct into a local minimum (Lin, 2007). Therefore, the alternating update rules

$$a_{iq} \leftarrow a_{iq} \frac{(XS^T)_{iq}}{(ASS^T)_{iq}} \quad \text{and} \quad s_{qj} \leftarrow s_{qj} \frac{(A^T X)_{qj}}{(A^T AS)_{qj}} \tag{4.4}$$

for $A$ and $S$ converge against a local minimum of (4.3) (Lee and Seung, 2001). The constraint in (4.3) is achieved by normalization of $a_i^q$ in each iteration (Cichocki et al., 2009). The multiplicative nature of this update rule guarantees a non-negative result if the initialization is non-negative. Since division by zero in (4.4) needs to be averted, usually a small positive constant is added to the denominator (Cichocki et al., 2009).

Despite the popularity of NMF the major drawback is that the multiplicative update rule (4.4) does not necessarily converge against the global solution (Donoho and Stodden, 2003). This can easily be seen by the fact that the found solution is not unique. In the simple case, one introduces a matrix $B$ such that $BB^T = I$. Then an equally valid solution is given by $\tilde{A} = AB$ and $\tilde{S} = B^T S$. The reason for this is that the alternative update rule is only convex in the individual steps but not as a whole (Lin, 2007). However, a multitude of initializations is found in the literature, see for example Cichocki et al. (2009).

### 4.3.2 Sparsity constraints for non-negative matrix factorization

While the original algorithm to solve the objective function (4.3) did not account for further constraints, it is now quite common to use such constraints as suggested by Hoyer (2004) and Daubechies et al. (2004). Even though factorization by (4.4) has the tendency to result in sparse solutions, it is often better to directly enforce them by specifying side conditions specific to the problem. Due to the intended interpretation of $S$ as characteristic spectra reflecting different metabolite and of $A$ as soft segmentation maps, both localized and sparse structures should be favoured. Hoyer (2004) introduced a sparsity measure

$$\text{sparseness}(x) = \frac{\sqrt{n} - \|x\|_1 / \|x\|_2}{\sqrt{n} - 1} \tag{4.5}$$

of a vector $x \in \mathbb{R}^n$ combining both $\ell_1$- and $\ell_2$-penalties. Accordingly, sparsity measures are incorporated in the computation of $A$ and $S$ by using the additive update rules

$$A \leftarrow A - \tau_A (AS - X) S^T \quad \text{and} \quad S \leftarrow S - \tau_S A^T (AS - X) \tag{4.6}$$

derived from gradient descent for some update steps $\tau_A, \tau_S$, followed by an orthogonal projection or thresholding to achieve the desired sparsity. However, the additive nature of (4.6) neither guarantees the non-negativity without thresholding nor results in a decrease of sparsity (4.5) between iteration steps (Hoyer, 2004).

Recently, it has been shown by Behrmann (2013) that the multiplicative update rule (4.4) can directly account for the functional

$$\min_{A,S} \frac{1}{2} \|X - AS\|_F^2 + \lambda_S \|S\|_1 + \frac{\mu_A}{2} \|A\|_F^2 + \frac{\mu_S}{2} \|S\|_F^2 \tag{4.7}$$

with additional constraints. In (4.7) the coefficients $\lambda_S$, $\mu_A$, and $\mu_S$ penalize the influence of these constraints, for a discussion see Bartels et al. (2013). A solution is found with a non-negative initialization of the matrices $A$ and $S$ and the following alternating iteration:

$$\begin{aligned} a_{iq} &\leftarrow a_{iq} \frac{(XS^T)_{iq}}{(A^T S S^T + \mu_A A)_{iq}} \\ s_{qj} &\leftarrow s_{qj} \frac{(A^T X)_{qj}}{(A^T A S + \mu_S S)_{qj} + \lambda_S} \end{aligned} \quad .$$

A derivation of this iteration as well as a discussion on its convergence properties can be found in literature (Hoyer, 2004; Lee and Seung, 2001). However, for this approach, the convexity problems described by Lin (2007) still hold.

### 4.3.3 Application to real data

In this section, the results by Kobarg et al. (2014) are presented where NMF with sparsity constraints (4.7) was applied to the rat brain data. The latent variable number estimation for PCA shown in Figure 4.4 suggest an underlying structure of 6 to 8 components. Here, $r = 6$ components were specified and the dataset was separated into regions strongly correlating with the anatomical features of the rat brain. The characteristic spectra (mixture signatures) computed by the algorithm also lack noise and fulfil the sparse signal approach. This way they are a more sophisticated approach than defining a dictionary of peak shapes, as the interconnection between different masses is preserved and provides greater detail for biological analysis. Hence, this approach clearly demonstrates, that

- Mixture signature $s_q, q = 1, \ldots, r$, can be determined by matrix factorization methods with sparsity constraints.
- Mixture component decomposition leads to data compression, i.e. instead of storing the full spectrum $x_i \in \mathbb{R}^m$, one only needs to store the $a_i \in \mathbb{R}^r$ (e.g. $r = 6$) coefficients of its approximate expansion with mixture signatures $S$ for the whole dataset.

Moreover, it should be emphasized, that computing a few characteristic spectra is the key for efficient analysis. However, in principle a full basis of characteristic spectra can be determined, i.e. allowing a full and exact recovery of each single measured spectrum as a weighted superposition of characteristic spectra. Each of these weighting coefficients (mixture coefficients), obtained form a partial or full decomposition, states the connection of the related characteristic spectrum for the given data.

The spatial plots of the weighting coefficients $a^q$ associated with the different mixture signatures $s_q$, $q = 1, \ldots, r$, are of particular interest for medical interpretation. These images provide a soft segmentation, which at least looks more natural than the usually used segmentation maps obtained by hierarchical clustering (Trede et al., 2012a; Watrous et al., 2011). The weighting coefficients $a_i^q$ state how strongly mixture signature $s_q$ contributes to the spectrum measured at position $i$. Hence, displaying $a^q$ as a spatial plot indicates regions having a molecular decomposition similar to the mixture signature $s_q$, see Figure 4.5.

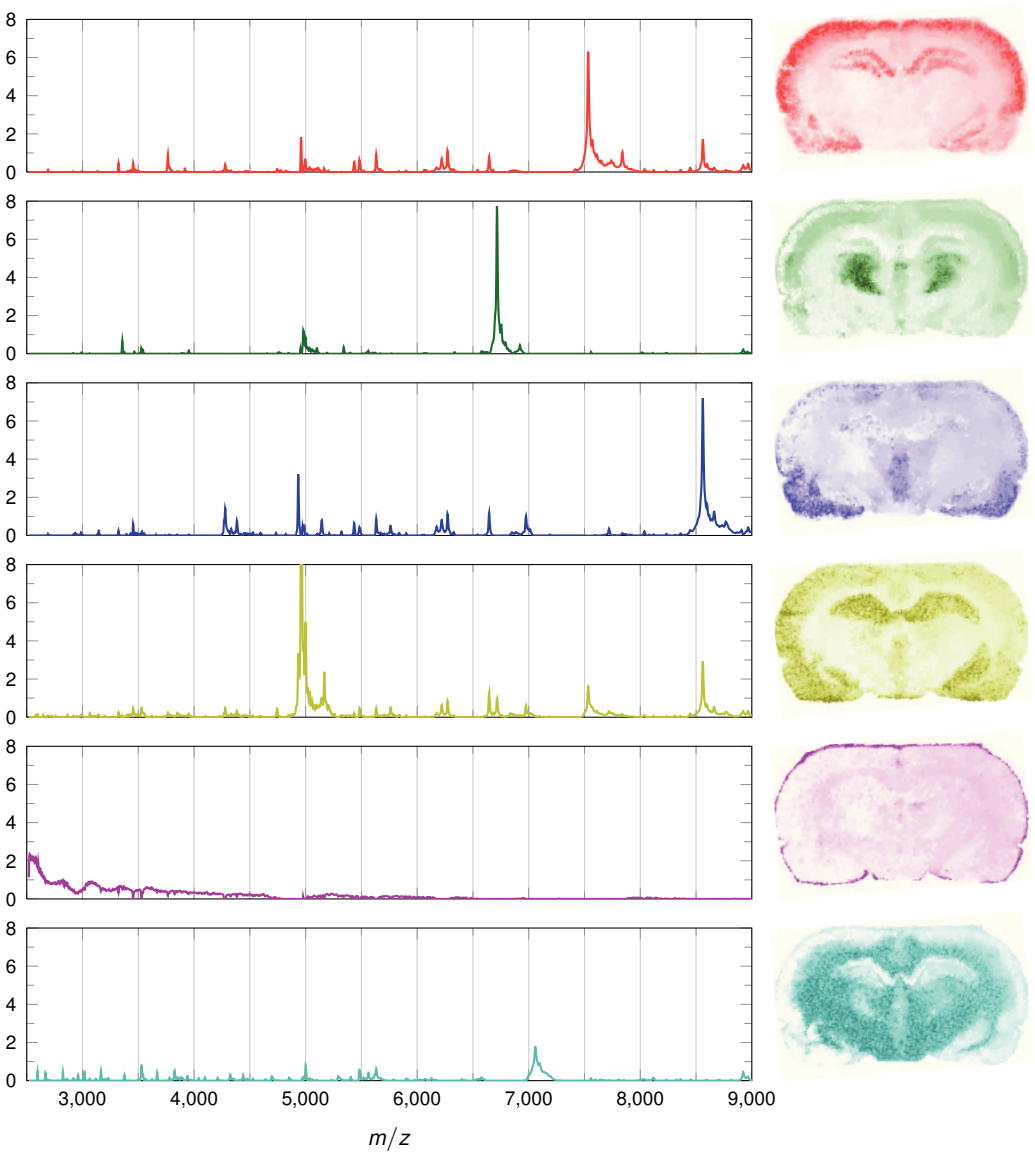



**Figure 4.5: Non-negative matrix factorization applied to the rat brain dataset.** Mixture signatures $s_q$ and images of weighting coefficient $a^q$ obtained by matrix factorization approach for $q = 1,\ldots,6$ in the rat brain dataset.

### 4.3.4 Discussion and future work

The spectrum model leads to the assumption, that the characteristic spectra reflect the different metabolic components present in the sample. Moreover, it is expected, that a medical expert can assign different metabolic functionalities with each of these spectra. This needs to be verified in future research projects in cooperation with medical experts. The score images are interpreted as a soft segmentation of the tissue section into regions with different metabolic structures. This interpretation is meaningful since the value of $a_i^q$ is the weighting coefficient of the $q$-th characteristic spectrum in the decomposition of the measured spectrum at location $i$. Hence, these images visualize the spatial relevance of the different characteristic spectra.

Analysing an individual mixture signature: This reveals characteristic spectral patterns, which are concentrated on one or several subintervals of the $m/z$ axis. Potential benefits are e.g. protein and biomarker identification, which no longer need to be based on selected $m/z$ values but can exploit the structure of characteristic spectra, which potentially resemble full isotope patterns for multiple molecular fragments.

This approach allows one to sequentially determine the most dominant metabolic structures of the available data. In the characteristic spectra determined from the experiment with the rat brain data, displayed in Figure 4.5, one can see that the characteristic spectra determine clusters (subintervals) on the $m/z$ axis which carry discriminative information.

Analysing the mixture coefficients related to the measured spectra: Another feature of the approach is the capability to recover each measured spectrum as a weighted superposition or mixture of the determined characteristic spectra. Analysing these mixture coefficients determines a measure of relevance, i.e. how strongly a chosen characteristic spectrum contributes to a measured dataset. For example in IMS experiments, these mixture coefficients can be visualized as a spatial plot revealing the characteristic metabolic structure of subregions. This is typically the basis for diagnosis or protein/biomarker identification.

#### Spatial constraints for non-negative matrix factorization

Additionally to the sparsity constraints in the previous section, spatial constraints can also be introduced. Since the coefficients in the matrix $A$ represent the pixel wise weights, these can be optimized in such a manner that they respect a certain spatial smoothness. Candidates that enforce spatial smoothness are the total variation norm or wavelet decomposition (Golbabaee et al., 2010). In that case, the matrix $A = \Psi_{2D}\Phi$ is itself described by a sparse representation of

coefficients $\Phi$ for a wavelet basis $\Psi_{2D}$. The theoretical background is presented exhaustively by Golbabaee et al. (2013). Using a wavelet basis to incorporate spatial relation is an approach also found in other fields such as the Bayesian image segmentation (Figueiredo, 2005).

**Accounting Poisson noise with Kullback-Leibler objective function**

Previously, the assumption for the model was $X \approx AS$, with the omission of an error term $E = X - AS$. The functional that was used assumed this error $E$ to be distributed normally (Cichocki et al., 2009; Sra and Dhillon, 2006). However, when the random experiment records counting events it is well known that the error need to be modelled by Poisson distribution (Keenan and Kotula, 2004). Correction for Poisson noise produces superior results when secondary ion mass spectrometry (SIMS) data is analysed with PCA (Wagner et al., 2006).

The original formulation of the NMF algorithm by Lee and Seung (1999) did account for Poisson noise, however, with a different functional instead of (4.3). The functional to minimize is chosen to be the log-likelihood

$$\sum_{i=1}^{n}\sum_{l=1}^{p}\left((AS)_{il} - x_{il}\log(AS)_{il}\right) \tag{4.8}$$

which is standard approach in Poisson data (Cichocki et al., 2009). Naturally, using (4.8) requires a different update rule. Following the same principle as before, the update rule is modified into

$$a_{iq} \leftarrow \frac{a_{iq}}{\|s_q\|_1}\sum_{l=1}^{p}\frac{s_{ql}x_{il}}{(AS)_{il}} \quad \text{and} \quad s_{ql} \leftarrow \frac{s_{ql}}{\|a^q\|_1}\sum_{i=1}^{n}\frac{a_{iq}x_{il}}{(AS)_{il}} \tag{4.9}$$

in this case (Lee and Seung, 2001). However, the update rules given in (4.9) will converge to a local minimum only (Lin, 2007) as it was the case for (4.4).

Interestingly, Gaussier and Goutte (2005) have claimed the assumption of Poisson noise and usage of (4.9) result in NMF becoming equivalent to probabilistic latent semantic analysis (PLSA). PLSA is gaining popularity in the MALDI field of IMS since the first use by Hanselmann et al. (2008). The theoretical properties of PLSA model a mixing of spectra via probability theory. However, there has been some debate both about the exact equivalence and need for such an extensive modification: While Gaussier and Goutte (2005) show the relation between the two algorithms, Ding et al. (2006) showed both algorithms will converge to different solutions even if initialized with the same values. Whether or not improved results for MALDI similar to SIMS (Wagner et al., 2006) can be achieved with any Poisson correction, is still open.

So far, Jones et al. (2012a) pointed out that each method is sufficient to identify major features in the data.

## 4.4 Distance-preserving projection of data with FastMap

Decomposing the data matrix into a coefficient/signature pair is not the only way to reduce the number of dimensions. However, if the goal is not visualize the data, but to do further computations, more efficient dimensionality reduction methods can be employed. For example, preserve the important properties, e.g. the distance between each spectrum. As discussed in Chapter 2.5, distances between the spectra are properties based on which a clustering can be obtained. Distance preserving projection can be achieved with multidimensional scaling (MDS, Hastie et al., 2009). However, the original distance matrix $D$ of size $n \times n$ is needed and an eigenvalue decomposition has to be computed. The FastMap algorithm (Faloutsos and Lin, 1995) is a related method and better suited in the context of IMS, as the computational cost is smaller. Compared to MDS, FastMap does not need the full distance matrix $D$, but only a small subset, so that implementation wise it can work on the dataset $X$ itself. This allows FastMap to be more memory preserving than MDS. Instead of $O(n^2)$ operations for computing the entire distance matrix in advance only $O(3n)$ computations per iteration are needed. The iterative projection of the high-dimensional data into a lower dimensional hyperspace returns new vectors. The distances between these new vectors are similar to the inter-distances originally present in the data.

Extensions to FastMap are also available, as already mentioned in the text, Wang et al. (2000) improved the result of FastMap in the case of non-Euclidean distances. Ostrouchov and Samatova (2005) addressed the point of selecting the pivot elements in each iteration. Objects of maximal distance between each other are often outliers. For the computation of the new axis system this is not always an optimal choice. However, authors showed a way to improve the quality of the mapping by using a robust selection strategy for the pivot elements. An improved pivot element strategy is also proposed by Ng and Huang (2002) where the goal is to obtain a well seperating segmentation of the data. The author's modified FastMap is used to decrease the amount of overlaps of data from different segments. This not only helps to segment the data more accurately, but also to visually discriminate the point clouds of the projected data.

Later in Section 5.3, FastMap will be used to provide an efficient implementation of spatial aware mapping (Alexandrov and Kobarg, 2011). However, the approach to find a coordinate

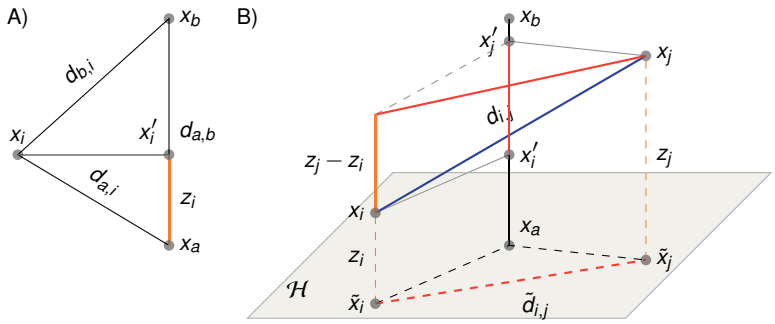



**Figure 4.6: Projection principles of FastMap.** A. Application of cosine law for projection onto new $x_a$-$x_b$-axis. B. Computation of new distance in $p-1$ dimensional hyperspace $\mathcal{H}$ from distance in $p$ dimensional space. (Reproduced from Faloutsos and Lin, 1995.)

representation given only a distance matrix can be further exploited. Here, two different applications will be introduced. First, colouring of segmentation maps is achieved with FastMap. Second, it will be used to display the similarity of $m/z$ images.

### 4.4.1 FastMap algorithm

The basic idea of FastMap is to use the two $p$ dimensional spectra $x_a$ and $x_b$ with greatest inter-distance $d_{a,b} = d(x_a, x_b)$ as pivot elements to form a new axis. A triangle can be formed with edges between $x_a$, $x_b$, and $x_i$ as shown in Figure 4.6A. The orthogonal projection of $x_i$ on the line of the $x_a$-$x_b$-axis divides the edge at the new point $x'_i$ in distance of $z_i = d(x_a, x'_i)$ from $x_a$. Within the two triangles Pythagoras' theorem gives

$$\begin{aligned} d_{b,i}^2 &= d_{i,i'}^2 + (d_{a,b} - z_i)^2 \\ &= d_{a,i}^2 - z_i^2 + d_{a,b}^2 - 2 z_i d_{a,b} + z_i^2 \end{aligned}$$

where solving for $z_i$ allows computing the scale

$$z_i = \frac{d_{a,i}^2 - d_{b,i}^2 + d_{a,b}^2}{2 d_{a,b}}, \quad i = 1, \ldots, n, \tag{4.10}$$

on the new axis. This shows, only the two rows $\{d(x_a, x_i)\}$, $\{d(x_b, x_i)\}$, and the inter-distance $d_{a,b}$ are the only parts needed of the distance matrix for (4.10). By design of the algorithm, these

two rows are even the same ones needed to find the pivot elements $x_a$ and $x_b$ and such they are not needed to be computed again.

Before proceeding to a new iteration, the spectra's projections $\tilde{x}_i$ into a $p-1$ dimensional hyperspace $\mathcal{H}$ are calculated. The hyperspace $\mathcal{H}$ is orthogonal to the $x_a$-$x_b$-axis as shown in Figure 4.6B. In this new iteration pairwise distances $\tilde{d}_{ij} = d(\tilde{x}_i, \tilde{x}_j)$ between the projected spectra $\tilde{x}_i$, $\tilde{x}_j$ in $\mathcal{H}$ will be needed. However, as $x_a$-$x_b$-axis is orthogonal to $\mathcal{H}$, Pythagoras' theorem can again be used and

$$\tilde{d}_{i,j}^2 = d_{i,j}^2 - (z_j - z_i)^2 \tag{4.11}$$

is obtained. Being dependent only on the scales for each spectrum, this again makes full computation of $D$ unnecessary. After finishing $q$ iterations of FastMap, the scales $z_{iv}$, $v = 1,\ldots,q$, correspond to the new coordinates for all mapped spectra $\tilde{x}_i = (z_{i1},\ldots,z_{iq})$, $i = 1,\ldots,n$.

The projections (4.10) and (4.11) are only valid for Euclidean distances. Wang et al. (1999) pointed out that in the case of non-Euclidean distances, negative squared distances might occur in (4.11), as $d_{i,j}^2 < (z_j - z_i)^2$ is possible. In this case, the sign of $d_{a,b}$ has to be preserved. A rather simple modification of equation (4.10) to

$$d_{a,b} = \text{sign}(d_{a,b}^2) \cdot \sqrt{|d_{a,b}^2|} \tag{4.12}$$

fixes this problem (Wang et al., 1999).

### 4.4.2 Assigning colours to segmentation maps with FastMap

Usually, the assignment of colours to segmentation maps is completely artificial and based on a label number. The label number for a segment depends on the initialization of $K$-means. This makes it harder to compare segmentation results for one dataset, as can be seen in Figure 4.7A)–D). One option is, to use accuracy measures based on (2.6) and (2.7) to align the new class labels to given ones. However, the choice of colour for the user-specified labels is still artificial in this case. While this overcomes the problem to assign a number of $K$ labels a colour map with $K$ colours, problems arise when the two segmentation maps to compare have different number of clusters and use different colours as well. It will be demonstrated, that using FastMap for determination of the segment colour is not artificial and even visually aids to capture the differences. A similar method has been developed by Guo et al. (2005) who use self-organizing maps and Fonville et al. (2013) who rely on PCA to achieve this automated colouring.

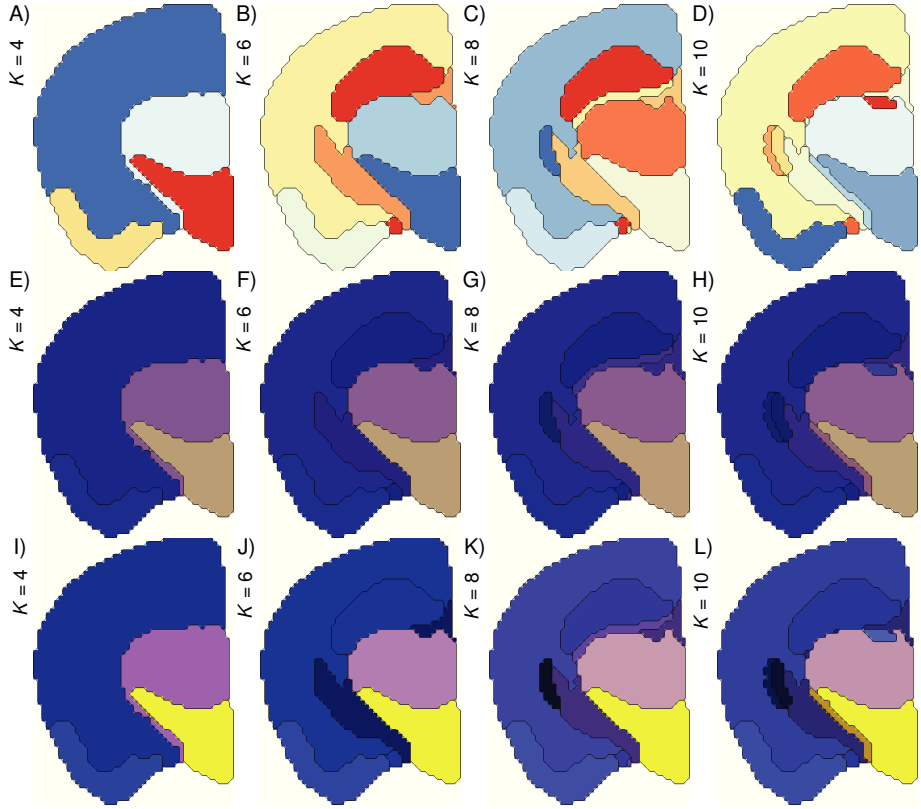



**Figure 4.7: Intercomparability of cluster results.** One section of the simulated dataset clustered by $K$-means with $K = 4, 6, \ldots, 10$ segments. A–D. Due to arbitrary segment numbering segment colours differ across results. E–H. FastMap was used to determine each pixel's RGB value and each segment in the image has received the RGB value of the segment's centroid. I–L. Contrast enhancement by maximizing the range of each RGB channel.

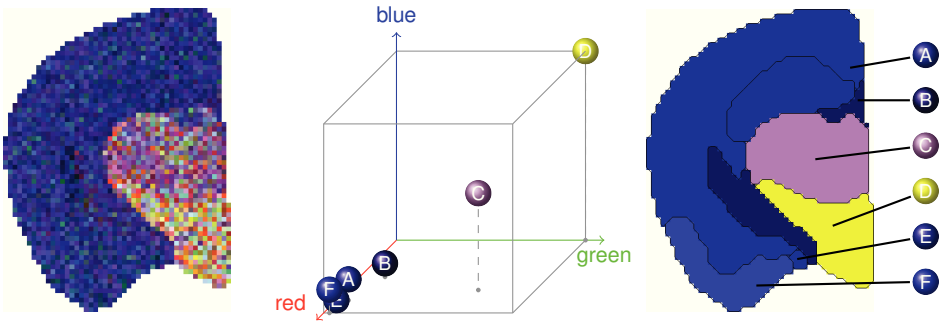



**Figure 4.8: Cluster centroids in 3D unit cube.** A. False colour image of data projected to RGB space. Each of the spectra in the dataset was projected to a $q = 3$ dimensional space, giving the pixels direct representation of similarity. B. RGB values of the segmentation's centroids in the unit cube. C. The corresponding segments with the new colour scheme.

First, for the entire dataset pivot elements need to be found. As before, these pivot elements define the coordinate system. Colours visible to the average human eye can be digitally represented by three colour channels. Since the aim is to give a colour representation, it is naturally to search for a FastMap space of dimension $q = 3$. The first thing one can do is to assign each spot individually by its representing RGB value. This representation in a RGB colour space already reveals the major features and spectral differences. Figure 4.8A) shows the major difference in spectral information represented by false colour in one single image.

This is not the case for segmentation maps, which only use a pseudo colour from a linear colour map. The colour of a pixel denotes its associated categorical class label only, but does not reveal the relation between the segments. However, segmentation maps can use an RGB colour space representation. For each segment one has to calculate the mean spectrum associated with this class. This can be done by calculating the average RGB value per class.

The approach proposed here can solve the problem of requiring a reference colourmap with exactly matching number of prespecified labels. Therefore, it is even better when the number of segments increases. It also helps to visualize segmentation results with different initializations, as demonstrated in Figure 4.7.

However, both for the segmentation maps in Figure 4.7E–H and the false colour image in Figure 4.8A a scaling of the RGB cube is advised as shown in 4.7I–L. Since FastMap uses two distinct spectra with great inter-distance, these are usually outliers. Efficient usage of the colour space therefore requires removal of the outliers by quantile thresholding. For the display of a series of segmentation maps, it is even required to rescale the RGB colour space to fit the

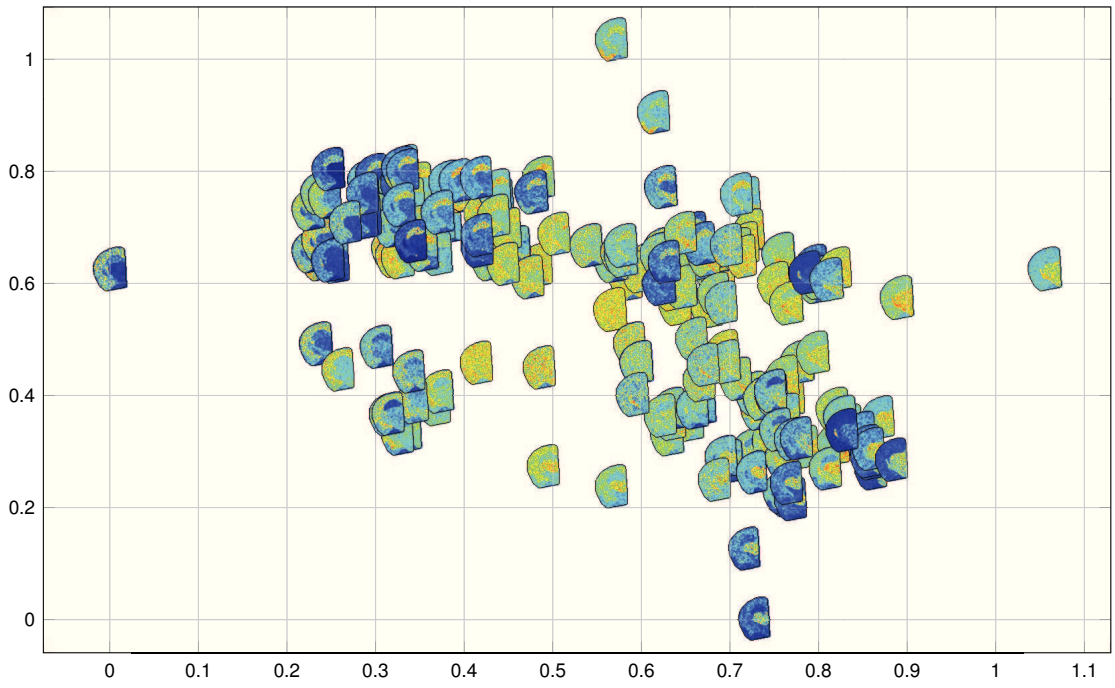



**Figure 4.9: Scatter plot of $m/z$ images at true peak positions.** Simulated $m/z$ images distributed in the plane where correlation distances between images are preserved.

centroids of the segmentations as shown in Figure 4.8. Otherwise the contrast is not high enough to discriminate the individual segments.

### 4.4.3 Displaying similarity of $m/z$ images with FastMap

As stated earlier, FastMap accepts a distance matrix as input and returns for each observation a set of points in Euclidean space for which the given distances are approximated. Therefore, it can be used to visualize relations between objects for which previously no ordinal scale was available. This is similar to producing a loadings plot, where the relative distances are visualized, with the difference of first introducing a new coordinate system.

Figure 4.9 shows the $m/z$ images of one of the coronal sections from the simulated dataset after preprocessing. The $m/z$ images are distributed in the plane according to their pairwise similarities with the relative similarities being preserved. In this case, similarity is defined by

the correlation value between each other. This is because the actual intensities are less important for the comparison. This image can only give an idea of the variability found in the data.

When the data is plotted, a correction for the use of the correlation measure is necessary. Since FastMap is designed for Euclidean distance the returned coordinate gives the relative position between the pivot elements. In cosine measure not only the sign correction (4.12) is needed, but also a compensation for most coordinates being near the origin of the axis system. This compensation is achieved before the display as a scatter plot, by using the inverse of the cosine.

### 4.4.4 Conclusion and related work

The work shown in this current section introduced the method of FastMap, where a distance preserving representation of the data is found. Furthermore, two applications were introduced. As it will be shown later on, FastMap is suitible to efficiently compute the spatial aware segmentation proposed by Alexandrov and Kobarg (2011).

One advantage of FastMap is the possibility to project more data into the same space defined by the pivot elements. In MDS this is not possible and recomputation of the entire process would be needed. As such, FastMap can be employed for database queries when a model is generated and later the representation in this reduced storage system is needed. Furthermore, this characteristic of FastMap can possibly be exploited, when the number of observed objects is too high to map the data in a single step. A small subset of the data should be sufficient to create the underlying axis system with pivot elements. All further objects can then be embedded in the same space. Especially, this is useful when the full data does not fit into the memory.

## 4.5 Summary of dimensionality reduction

In this chapter several dimensionality reduction methods have been presented. Due to the nature of the data, some sort of dimensionality reduction is a must to efficiently perform further computational analysis. The list of dimensionality reduction methods presented in this chapter is by no means exhaustive. Several other approaches that were not covered in this chapter have been proposed in the context of mass spectrometry (Lee and Gilmore, 2009; Jones et al., 2012a).

As it has been shown, PCA is a powerful tool to discover information hidden in data. However, in the context of IMS, the interpretability of the resulting spectra is lacking intuitive interpretation. Furthermore, the computational complexity and the noise level in the data prevent its application to the full dataset. Before more efficient algorithms were available, performing PCA

was one of the few possibilities to run clustering algorithms for IMS data (McCombie et al., 2005; Deininger et al., 2008). However, more advanced methods such as the presented NMF are easy to interpret as the full spectrum is used. The resulting spectra are not artificially reduced to line spectra as in the case of PCA, but are found due to penalties Lee and Seung (1999); Hoyer (2004). Additionally, the score images are constructed with the side condition to use additive mixture only. This creates a soft-segmentation which is easily to comprehend. Because of the flexibility of the methods used in NMF, several goals can be pursued in future research. Aside from the matrix factorization approach for dimensionality reduction, the completely different concept of distance preserving mapping with FastMap was described (Faloutsos and Lin, 1995). FastMap will be used in the second part of the following chapter to efficiently compute segmentation maps as was originally proposed by Alexandrov and Kobarg (2011). In the present chapter, further applications of FastMap in the context of data analysis have been introduced.

PCA will keep its reputation as a primary analysis tool, while alternatives such as NMF or FastMap only slowly gain popularity (Jones et al., 2012a). However, since dimensionality reduction is such an important topic in wide areas of application, different scientific communities try different approaches. In future, analysis of large IMS datasets can benefit from methods recently developed. For example, principles from compressed sensing have been successfully transferred by Bartels et al. (2013) in order to solve bigger problems with reduced computational effort. Randomly generated basis systems also seem to generate promising results in general (Mahoney and Drineas, 2009) and specifically for IMS (Palmer et al., 2013).

# 5 Noise reduction methods

## 5.1 Motivation for this chapter

After the previous chapter has introduced the concept of reducing the number of image channels, this chapter will focus on noise reduction methods. Two types of noise reduction are considered in imaging mass spectrometry (IMS). The first is the traditional denoising of the individual spectra with either the Savitzky-Golay filter (Savitzky and Golay, 1964; Källback et al., 2012; Urban et al., 2014) or wavelet-based methods (Lange et al., 2006; Kwon et al., 2008; Morris et al., 2008; Shin et al., 2010). The approach is often used improve retrieval of the original peak shapes from the data. In the case of mass spectrometry where few spectra are observed this is practicable. Furthermore, smoothing in the direction of the mass is the only way to counteract the distortion by ions belonging to the same mass but arriving at different times at the detector. However, especially in the case of imaging data, spatial information is also available and can be used during noise removal (McDonnell et al., 2008; Alexandrov et al., 2010; Bruand et al., 2011a). The idea behind spatial smoothing is that pixels which are spatially close to each other

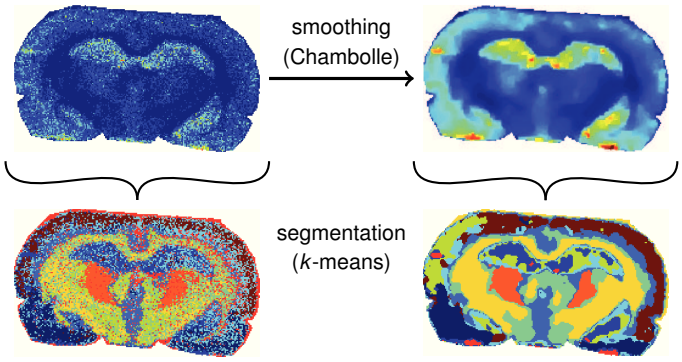



**Figure 5.1: Effect of edge preserving spatial smoothing on the segmentation result.** Without denoising the segmentation map has a noisy appearance, while it clearly shows the structure after smoothing the data.

in general have similar spectra. For the IMS data the positive effect of spatial smoothing can be seen in the segmentation maps shown in Figure 5.1. Direct application of a clustering algorithm results in a pixelated segmentation map, while the segments are connected when clustering is applied to smoothed data.

The first part of the chapter will introduce the spatial smoothing by Chambolle (2004). Chambolle (2004) used an iterative process to obtain a smoothed image, while preserving edges. Preserving the edges in an image is a desirable objective, especially in IMS, where it is expected that these divide different anatomical features. This approach will be compared against the performance of a filter based smoothing. Filter based smoothing can be carried out in a fast manner, and has the potential to be parallelizable.

The second half of this chapter will use a different approach to obtain the spatial smoothing effect. Spatial smoothing can also be achieved by embedding the data into a feature space (Alexandrov and Kobarg, 2011). The feature space is of a higher dimension such that the segmentation by linear hyperplanes becomes highly effective. Conceptionally, the approach is inspired by the kernel trick from support vector machines (Hastie et al., 2009; Luts et al., 2010; Tarabalka et al., 2010). Segmentation has the goal to find a linear separation between existing groups in the dataset. Rather than use segmentation algorithms to find non-linear separating hyperplane in the dataset, the input data is transformed by the kernel function. Here, the kernel is defined by the Gaussian weights with the result that linear separation can be applied. Since the mapping function produces a higher dimensional space than the input data, the dimensionality reduction method FastMap from the previous chapter is employed. Usage of FastMap is computationally less expensive and requires less memory. This helps to accelerate the final step of obtaining the segmentation map.

## 5.2 Channel-by-channel spatial smoothing

When the individual image channels are denoised separately, one also speaks of *channel-by-channel spatial smoothing*. As it is already demonstrated in Figure 5.1, spatial smoothing has a positive effect on segmentation results. However, not all denoising strategies are applicable: simple denoising strategies are usually not able to preserve edges between intensity regions. Since the general idea of denoising is to remove abrupt changes of intensity, this assumption is not valid where two different regions meet in the image. A solution can be in using the so called *edge-preserving denoising* as introduced by Alexandrov et al. (2010).

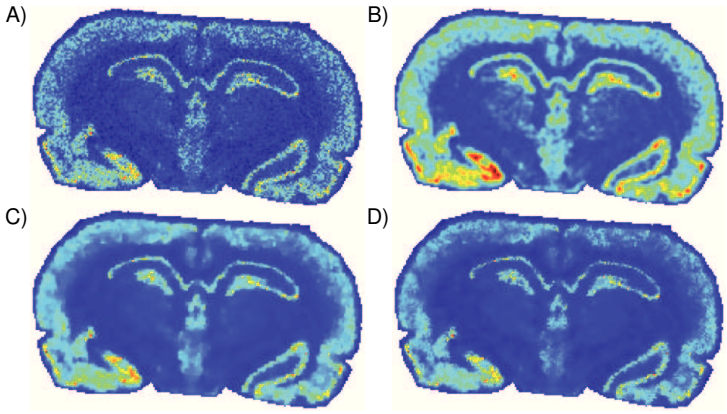



**Figure 5.2: Overview of different noise reduction methods.** A. Original image $m/z$ 6223 from rat brain dataset. B. Smoothed with Gaussian filter, $w = 3, \sigma = 1.5$. C. Smoothed by iterative Chambolle, $\lambda = 10$. D. Smoothed by bilateral filter, $w = 3, \sigma = 1.5, \lambda = \frac{1}{8}$.

The main focus will be on algorithms that compensate the effect of smoothing regions with different intensity. Alexandrov et al. (2010) used a computationally expensive variant proposed by Grasmair (2009) which is an iterative denoising algorithm. It has the advantage that it is adaptive, meaning that the smoothing effect of an image is described by a locally dependent parameter. Trede et al. (2012b) use the method introduced by Chambolle (2004) which only has a global paramter independent from the spatial position. An alternative is to use the so called bilateral filter by Tomasi and Manduchi (1998). Which has a similar adaptive approach, but is computationally less expensive, since ordinary filter functions are used (Chen et al., 2007).

Figure 5.2 shows the different outcomes. In comparison to an unsmoothed $m/z$ image, a Gaussian filter, Chambolle's algorithm, and a bilateral filter are shown. Note that for the Gaussian filter, the intensity regions grew significantly, while both results for Chambolle and bilateral filter are closer to the original structure, but preserved local structures.

Let $h : \Omega \to \mathbb{R}$ be an image defined over a support $\Omega \subset \mathbb{N}^d$ that maps each coordinate into the colour space $\mathbb{R}$. Note that in the following for simplicity $\Omega$ is here assumed as the $d = 2$ dimensional discrete support, but can easily be extended to $d = 3$. Likewise, the number of colour channels is set to one as each channel $l = 1, \ldots, p$ is treated separately. For example, with $\Omega = \{1, \ldots, n_1\} \times \{1, \ldots, n_2\}$ the image $h$ is an element in the Euclidean space $\mathcal{H} = \mathbb{R}^{n_1 \times n_2}$.

### 5.2.1 Chambolle's algorithm

Spatial smoothing of an image while preserving some structure found in the image can be achieved with the method

$$\min_{g \in \mathcal{H}} \frac{1}{2\lambda} \|g-h\|_2^2 + J(g) \tag{5.1}$$

proposed by Rudin et al. (1992). In this setting, given a noisy image $h : \Omega \to \mathbb{R}$, the goal is to find a minimizer of (5.1), where

$$J(g) = \int_\Omega |\nabla g(\omega)| \, \mathrm{d}\omega$$

is called the *total variation* of $g$, computed by the integration of the absolute value of the gradient $\nabla g : \Omega \to \mathbb{R}^d$, $\nabla g \in \mathcal{G} = \mathcal{H} \times \mathcal{H}$. Chambolle (2004) has shown that the solution to (5.1) can be equally obtained by the dual problem

$$\min_{|p|<1} \|\operatorname{div} p - \tfrac{h}{\lambda}\|_2^2 \tag{5.2}$$

where $g = h - \lambda \operatorname{div} p$. The dual problem (5.2) is solvable by

$$-\nabla(\operatorname{div} p - \tfrac{h}{\lambda}) + \alpha p = 0$$

via the Karush-Kuhn-Tucker conditions (Hiriart-Urruty and Lemaréchal, 1996). Case-by-case analysis with $\alpha = |\phi|$ where $\phi = \nabla(\operatorname{div} p - \frac{h}{\lambda})$ gives $-\phi + |\phi| p = 0$ which can be solved by fix point iteration $p = p + \tau(\phi - |\phi| p)$ the update rule giving

$$p^{(c+1)} = \frac{p^{(c)} + \tau \phi^{(c)}}{1 + \tau |\phi^{(c)}|}$$

for the problem. A proof for the convergence of this update rule is given by Chambolle (2004). The same paper also shows how the discretization of divergence and gradient have to be computed in order to apply the method to digital images.

### 5.2.2 Bilateral filtering

For the smoothing of an image, the most simple approach is to use a convolution filter $\phi$. To obtain a smoothed version $g$ of the image $h$ one calculates

$$g(\xi) = \frac{1}{\Phi(\xi)} \int_\Omega h(\xi)\phi(\xi,\omega)\,\mathrm{d}\omega$$

for each pixel $\xi \in \Omega$ where

$$\Phi(\xi) = \int_\Omega \phi(\xi,\omega)\,\mathrm{d}\omega$$

is the local normalization. A well known filter is the Gaussian filter where

$$\phi(\xi,\omega;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp - \frac{\|\xi-\omega\|_2^2}{2\sigma^2} \tag{5.3}$$

and $\sigma$ a parameter influencing the smoothing effect. This special filter is shift invariant and therefore depending on the distance $r = \|\xi-\omega\|_2$ only. The values of $\phi(r;\sigma)$ decrease with distance $r$ from the central pixel and are close to zero for $r > 1.96\sigma$ which is known as the *two sigma rule* (Upton and Cook, 2008). Therefore, the filter is usually defined on a small window of size $w \times w$, with $w = 2r+1$, only allowing fast computation. Smoothing an image with this filter removes noise, but also introduces a blur to certain types of images (Bredies and Lorenz, 2011).

In bilateral filtering by Tomasi and Manduchi (1998) the principle is extended to also weight the colour intensity with

$$\psi(\xi,\omega;\lambda) = \frac{1}{\sqrt{2\pi}\lambda} \exp - \frac{\|h(\xi)-h(\omega)\|_2^2}{2\lambda^2} \tag{5.4}$$

as a filter function constructed with the same principle as (5.3). This acts as a measure of difference of intensities. However, the smoothing should still be local. A locally smoothing filter with adaptive weights (5.4) is constructed by the combination of the filters and gives

$$g(\xi) = \frac{1}{\Phi(\xi)\Psi(\xi)} \int_\Omega h(\xi)\phi(\xi,\omega)\psi(\xi,\omega)\,\mathrm{d}\omega$$

as the smoothed version of the input image $h$. This way differing intensities are compensated leading to an edge preserving smoothing filter (Tomasi and Manduchi, 1998).

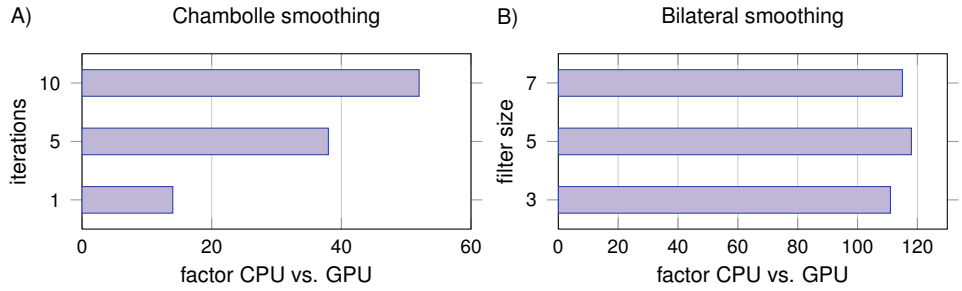



**Figure 5.3: Runtime comparison of smoothing algorithms.** Pure runtime acceleration factors of Chambolle's algorithm and bilateral filtering. Each algorithm is tested for a MATLAB/MEX framework vs. GPU implementation.

### 5.2.3 Comparison of channel-by-channel spatial smoothing

To compare the two channel-by-channel smoothing methods, several numerical experiments were performed. First it has to be evaluated if the transfer of the algorithm to GPU produces the same numerical results as on the CPU. From this benchmark experiment also the gain of speed can be computed. Such benchmarking was performed within the UNLocX project for which SagivTech Ltd. produced GPU code and ensured the numerical stability. The results from the corresponding work package report (Kobarg et al., 2012) are summarized after the description of the experiment setup. The second phase of evaluation can then compare the smoothing methods directly. Since the smoothed image channels are expected to be different by the nature of the smoothing techniques, the segmentation maps obtained by $K$-means after smoothing are used as comparison criterion. Again, the results are taken from the UNLocX report (Kobarg and Maass, 2013). Additionally to the research within the UNLocX project, in this thesis a strategy for choosing good parameters is demonstrated. A single section of the simulated data is used where the label of each pixel is known. With this, a direct comparison of the segmentation map obtained after smoothing and the ground truth can be performed.

For the benchmark experiment, the first 2048 image channels of the rat brain dataset are used. The numerical difference between the results of the implementations is only affected by the machine precision. A GPU only works with single precision, where CPU can handle double precision. With the confirmation both implementations work identically, runtime comparisons are performed. These experiments have the goal to estimate how much speed improvement can be gained by transferring the work from a pure CPU environment to a GPU environment. As expected, the speed gain of bilateral filter is much higher in comparison to the one of Chambolle,

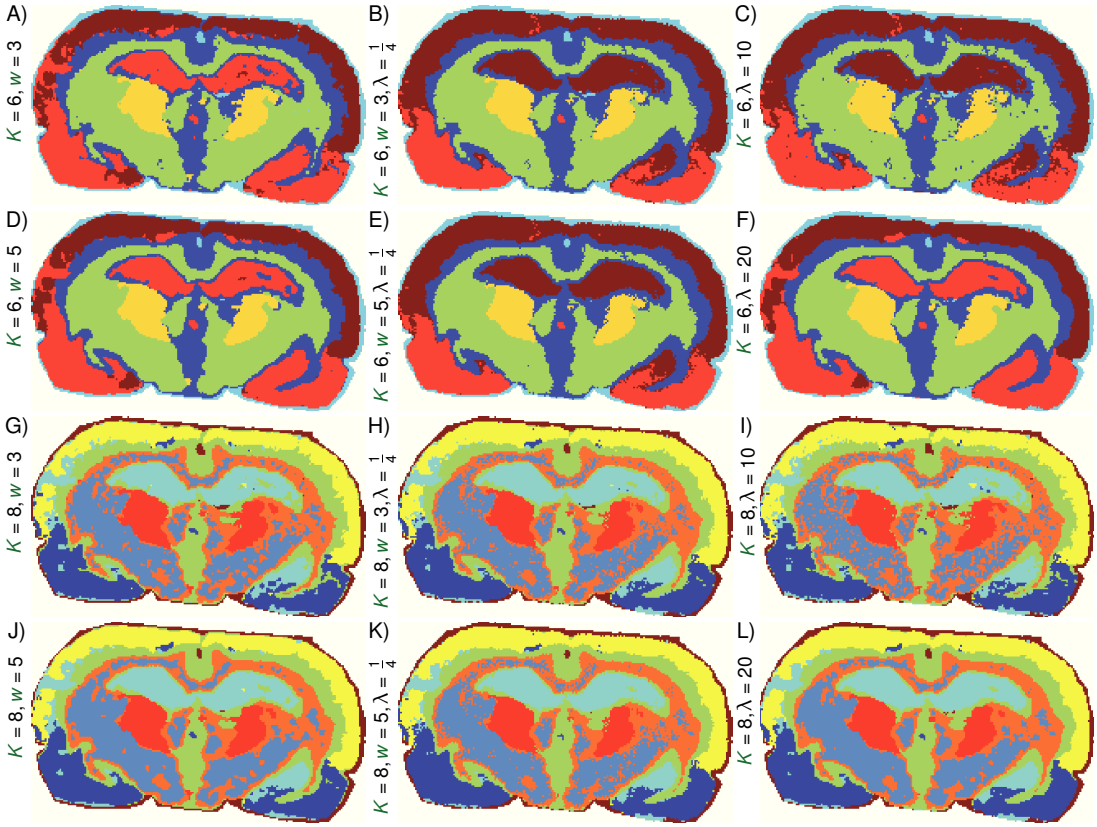



**Figure 5.4: Segmentation maps depending on channel-by-channel smoothing method.** Obtained with $K$-means for rat brain data (71 peaks) after spatial smoothing with Gaussian, bilateral, Chambolle algorithm (from left to right). A–C. Weak smoothing and 6 classes (filter size $w = 3$ resp. $\lambda = 10$). D–F. Medium smoothing and 6 classes (filter size $w = 5$ resp. $\lambda = 20$). G–I. Weak smoothing and 8 classes. J–L. Medium smoothing and 8 classes.

(a) 6 classes, weak

| | gauss | bilat. | chambolle |
|---|---|---|---|
| gauss | 1 | 0.8971 | 0.8903 |
| bilat. | 0.8971 | 1 | 0.9773 |
| chambolle | 0.8903 | 0.9773 | 1 |

(b) 6 classes, medium

| | gauss | bilat. | chambolle |
|---|---|---|---|
| gauss | 1 | 0.9107 | 0.9789 |
| bilat. | 0.9107 | 1 | 0.9178 |
| chambolle | 0.9789 | 0.9178 | 1 |

(c) 8 classes, weak

| | gauss | bilat. | chambolle |
|---|---|---|---|
| gauss | 1 | 0.9436 | 0.9405 |
| bilat. | 0.9436 | 1 | 0.9646 |
| chambolle | 0.9405 | 0.9646 | 1 |

(d) 8 classes, medium

| | gauss | bilat. | chambolle |
|---|---|---|---|
| gauss | 1 | 0.9516 | 0.9637 |
| bilat. | 0.9516 | 1 | 0.9744 |
| chambolle | 0.9637 | 0.9744 | 1 |

**Table 5.1: Accuracy of different spatial smoothing settings.** Pairwise balanced accuracy between segmentation maps for rat brain dataset.

as it can be clearly seen in Figure 5.3. Since Chambolle is an iterative algorithm this is the major factor influencing the runtime. Apparently, the runtime of the bilateral filter is independent from the filter size $w$. Bilateral filter is known to be a GPU friendly algorithm that is highly parallelizable.

The second open question concerns the difference between the segmentation maps that are obtained when the smoothing method is changed. Since both methods work differently, a direct comparison of the smoothed channel images is not possible. The goal in the processing pipeline is also not related to retrieve unsmoothed images, but characteristic $m/z$ values for segments detected by clustering. $K$-means is used with a predefined set of $K = 6$ and $K = 8$ classes. Beside the filters presented in this section, also a standard Gaussian filter is used for comparison. Two settings of smoothing are compared, first the filter based smoothing utilized a filter size $w = 3$ for weak smoothing and $w = 5$ for medium smoothing. For Chambolle the parameters $\lambda = 10$ and $\lambda = 20$ are used for weak and medium smoothing. The results in Figure 5.4 show, the segmentation maps are similar in general. The majority of pixels is grouped together across the results. In the case of $K = 6$ the part of the hippocampus region is identified as an own cluster in some of the results. For weak smoothing, both bilateral and Chambolle identify it as an independent cluster, while for medium smoothing only bilateral creates this cluster. The region of hypothalamus appears to be less consistent in the Chambolle result for $K = 8$. The level of noise within this segment is already fine for the weak setting in both the Gaussian and the

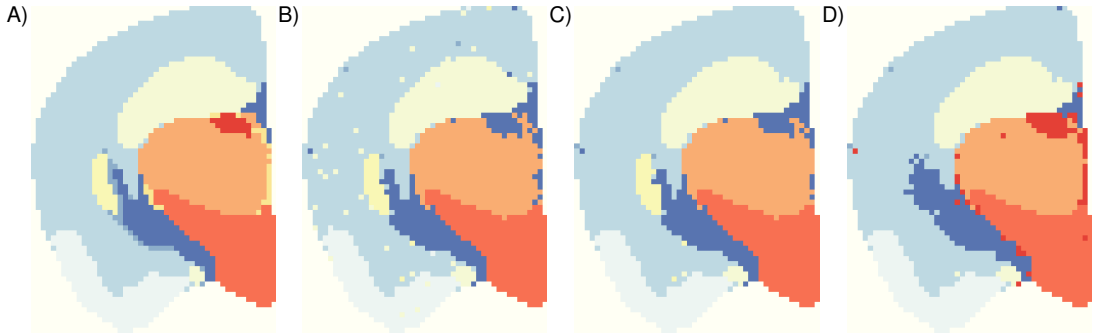



**Figure 5.5: Segmentation maps of best parameters for bilateral filtering.** Results for a single section generated with the simulation framework at 100 µm resolution with 196 peaks and strong noise. A. Ground truth of simulated data. B. Weak smoothing, parameters $w = 3$ $\sigma = 2.65$ and $\lambda = 0.07$. C. Medium smoothing, parameters $w = 5$ $\sigma = 2.4$ and $\lambda = 0.11$. D. Strong smoothing, parameters $w = 7$ $\sigma = 1.6$ and $\lambda = 0.12$.

of hypothalamus appears to be less consistent in the Chambolle result for $K = 8$. The level of noise within this segment is already fine for the weak setting in both the Gaussian and the bilateral. The scores for cluster comparison shown in Table 5.1 confirm high similarity between the results. More 89.03 % of the labels are identical in the weak smoothing case when compared against Gaussian filter and the similarity between bilateral filter and Chambolle is even 97.73 %. A likewise high range of similarity is achieved for the medium smoothing and $K = 6$ classes, here the result is varying because of bilateral filter creating the individual hippocampus region. For the $K = 8$ case the range is smaller starting from 94.05 % and going up to 97.44 %.

The experiment from UNLocX has shown, the segmentation map after clustering is sensitive to the parameters used in the smoothing of the data. Furthermore, bilateral filter has one extra parameter that controls the smoothing. To understand the effect of the parameter in bilateral filtering, the following experiment is performed with the simulated data. The segmentation maps obtained after smoothing are compared against the known labels for each pixel. For a defined range each parameter pair of the image channels of the ground truth peaks, the data is smoothed by bilateral filtering. Using the ground truth peaks averts introducing errors by omitting important peaks. The results in Figure 5.5 show the best matching segmentation map. In total there are 10 classes present in the given section. However, $K$-means with $K = 8$ classes produces more accurate segmentation maps, by dropping regions with few pixels. for $K = 10$ classes are split into smaller segments rather than identifying the small regions. Thereofre, the only the results for $K = 8$ are shown here.

### 5.2.4 Discussion of channel-by-channel smoothing

The algorithms that were presented show similar results. One advantage of both the bilateral as well as the Gaussian filter are their high potential to be parallelizable, while Chambolle is based on iterations.

From the experiment shown in this section it can be derived, that the use of bilateral filter weights produces smooth segmentation maps. Both in simplicity and speed gain, the bilateral filter outperforms the method proposed by Chambolle (2004). In the next section, a derivation of the method will be presented. The filter weights will be used to prepare the data directly for segmentation.

It should be noted, that both algorithms have been extended to be used on multi-channel images ($p = 3$). The projection algorithm of Chambolle was extended to account for colour images by Duran et al. (2013) and the bilateral filter with a vector norm instead of the absolute value was proposed already by Tomasi and Manduchi (1998). However, in the case of the bilateral filter, a direct transfer to RGB colours proved to be difficult as artefacts were created due to the mixing of the three base colours. When smoothed with the bilateral filter, edges between colours are preserved, but a colour shift does occur. For RGB colour images this shift can be avoided by transforming the images to the L*a*b colour space where the colour shift does not occur. However, when $p > 3$ with multiple colour channels are used such as in IMS there is no known transfer. Therefore, the spatial smoothing should be carried out in a channel-wise manner.

## 5.3 Spatially aware segmentation

In this section, efficient noise-suppressing segmentation based on a spatially aware embedding approach is presented. This merges denoising and dimensionality reduction into one step as shown in Figure 5.6. The embedding function $\Phi$ is defined based on a window of $w \times w$ pixels and weights $\{\alpha_{ij}\}$. The embedding function and the weights are used to project $n$ spectra of length $p$ into a high dimensional feature space of dimension $pw^2$, as demonstrated in Figure 5.7. The points in the feature space can be processed with standard clustering algorithms. Furthermore, it will be shown how to apply the efficient dimensionality reduction algorithm *FastMap* (Faloutsos and Lin, 1995) to speed up the procedure and to reduce the memory requirements. Moreover, the basic principle of the mapping strategy is extended to make the procedure edge-preserving. The concept of embedding the spatial information in the data was reported earlier

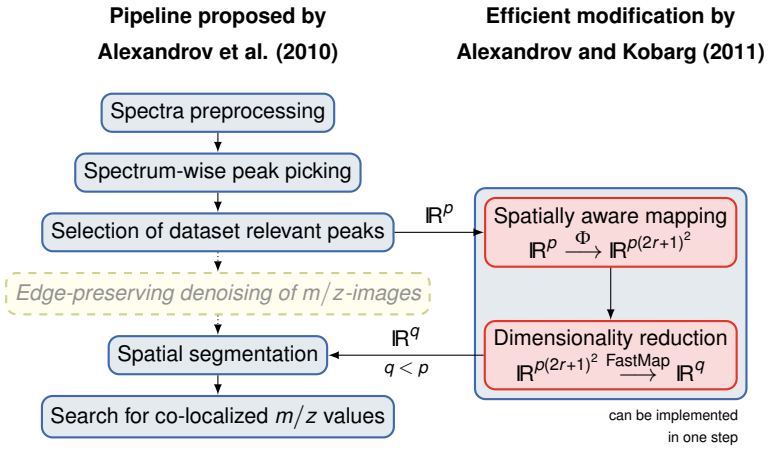



**Figure 5.6: Chart of efficient processing pipeline.** The individual steps of the efficient modification for the segmentation pipeline as proposed by Alexandrov and Kobarg (2011).

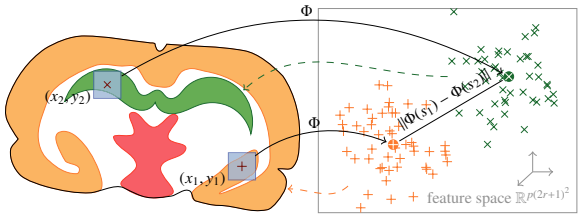



**Figure 5.7: Spatial embedding into feature space.** Transformation of spectra $s_1$ and $s_2$ into feature space by taking neighbourhoods around $(x_1,y_1)$ and $(x_2,y_2)$ into account; the mapped positions $(x_1,y_1)$ and $(x_2,y_2)$ are clustered in feature space. (Reproduced from Alexandrov and Kobarg, 2011.)

(Alexandrov and Kobarg, 2011), where it was applied for segmentation of 2D MALDI IMS data. It was also extended to 3D and applied to a simulated 3D MALDI IMS dataset (Kobarg and Alexandrov, 2013). The following section is mainly based on these two publications. The method was also applied to obtain segmentation maps in non-biological research (Bemis et al., 2012) and was adapted into an open source software package (Bemis et al., 2013).

### 5.3.1 Weighted embedding into feature space

Probably the most apparent way to embed the spatial relations between pixels into a clustering algorithm is to use a distance-based clustering for the distance $d(s_1,s_2)$ between two spectra $s_1$ and $s_2$ measured at spatial coordinates $(x_1,y_1)$ and $(x_2,y_2)$ respectively. With distance-based

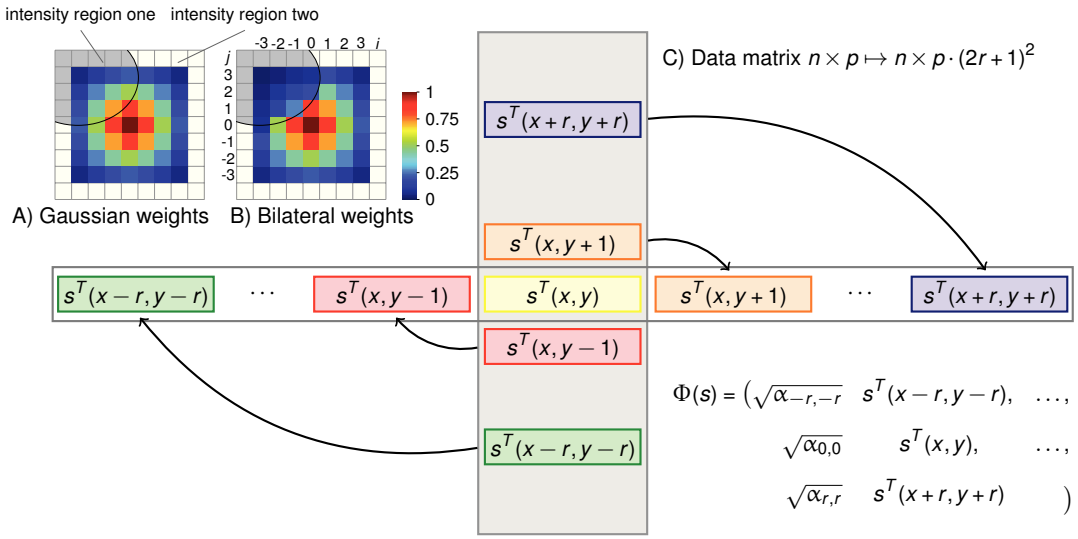



**Figure 5.8: Principle of concatenation of spectra as a form of data mapping.** The data matrix originally contains $n \times p$ spectra for which spatial positions $(x,y)$ are known. The intensity vectors relative to the spatial coordinate are placed accordingly to the offset. This increases the data matrix to $n \times p \cdot (2r+1)^2$. The mapping itself adds filter weights for based on the spatial distance (SA case) and intensity difference (SASA case). (Reproduced from Alexandrov and Kobarg, 2011.)

clustering it is easy to replace the default distance function, such as the Euclidean distance $d(s_1,s_2) = \|s_1 - s_2\|_2$ between the intensity vectors, by one working with a filter window of width $w = 2r+1$. A distance function, that uses information from neighbouring spectra in small window of radius $r$ then looks like

$$d_{r,\{\alpha_{ij}\}}(s_1,s_2)^2 = \sum_{-r \le i,j \le r} \alpha_{ij} \|s(x_1+i,y_1+j) - s(x_2+i,y_2+j)\|_2^2 , \tag{5.5}$$

where $\{\alpha_{ij}\}$ are factors weighting the influence of pixels from the neighbourhood. It is natural to choose weights $\{\alpha_{ij}\}$ which decrease with increasing $i^2+j^2$. For pixels distant from the neighbourhood centre the weights will be small. In a neighbourhood of radius $r$, the *Gaussian weights* are defined as

$$\alpha_{ij} = \exp \frac{-i^2-j^2}{2\sigma^2}, \tag{5.6}$$

with $\sigma = (2r+1)/4$ selected according to the two-sigma rule (Upton and Cook, 2008), see Figure 5.8A.

Unfortunately, this approach is both time and memory-consuming for datasets with many spectra, since it requires calculating a distance matrix of size $(n^2-n)/2$ and storage space for each of those values. Therefore, the spectra of length $p$ are mapped into a Euclidean feature space $\mathcal{F}$ using a mapping $\Phi\colon \mathbb{R}^p \to \mathcal{F}$. The feature space is selected such that within $\mathcal{F}$ the standard Euclidean distance

$$\|\Phi(s_1)-\Phi(s_2)\|_2 = d_{r,\{\alpha_{ij}\}}(s_1,s_2)$$

equals the desired distance (5.5) as illustrated in Figure 5.7. This can be achieved by using the mapping

$$\begin{aligned}\Phi(s) = \Phi(s(x,y)) = \big[&\sqrt{\alpha_{-r,-r}}\, s^{\mathrm{T}}(x-r,y-r),\ldots,\\ &\sqrt{\alpha_{0,0}}\, s^{\mathrm{T}}(x,y),\ldots,\sqrt{\alpha_{r,r}}\, s^{\mathrm{T}}(x+r,y+r)\big]^{\mathrm{T}},\end{aligned} \tag{5.7}$$

which describes the concatenation of spectra $s(x+i,y+j)\in\mathbb{R}^p$, $i,j=-r,\ldots,r$, in the neighbourhood of spectrum $s(x,y)$ to one single vector, as shown in Figure 5.8C. Each neighbouring spectrum is multiplied by a square root of the corresponding weight. Naturally, the feature space $\mathcal{F}$ is $\mathbb{R}^{pw^2}$ for such $\Phi$. If $n \gg p$ and $r$ is small, then storing the mapped data of size $n\times pw^2$ is significantly cheaper than $(n^2-n)/2$ pairwise distances.

### 5.3.2 Structure adaptive weights

Applying the spatially-aware clustering proposed in the previous section to IMS data, shows that it improves the segmentation maps as compared to the straightforward clustering. However, it can smooth the edges between the anatomical or histological regions or eliminates small details; more on this will be discussed in the next section. Use of a smaller pixel neighbourhood radius $r$ solves this problem only partially, since for a smaller $r$ the noise in the resulted segmentation map is stronger.

So, another method is proposed by Alexandrov and Kobarg (2011), where for a pixel, the weights of pixels in its neighbourhood are not simply Gaussian (5.6) but calculated adaptively, taking into account similarities of the pixels. The key idea is borrowed from the bilateral filtering (Tomasi and Manduchi, 1998) where the adaptively calculated weights are used afterwards for averaging. For each pixel in the neighbourhood, the distance between its spectrum and the

spectrum in the centre of the neighbourhood is considered. The larger the distance (the less similar the spectra are), the smaller the weight.

First, for a pixel with coordinates $(x,y)$,

$$\beta_{ij}(x,y) = \exp\frac{-\delta_{ij}(x,y)^2}{2\lambda^2}, \quad -r \le i,j \le r, \tag{5.8}$$

is introduced in line with Tomasi and Manduchi (1998), where $\lambda$ is a parameter and $\delta_{ij}(x,y) = \|s(x+i,y+j) - s(x,y)\|_2$. Then the distance between two spectra at coordinates $(x_1,y_1)$ and $(x_2,y_2)$ is as follows

$$d_{r,\{\tilde{\alpha}_{ij}\},\lambda}(s_1,s_2)^2 = \sum_{-r\le i,j\le r} \tilde{\alpha}_{ij}(x,y)\|s(x_1+i,y_1+j) - s(x_2+i,y_2+j)\|_2^2, \tag{5.9}$$

$$\tilde{\alpha}_{ij}(x,y) = \alpha_{ij}\sqrt{\beta_{ij}(x_1,y_2)\beta_{ij}(x_2,y_2)},$$

which differs from (5.5) by using the adaptive weights $\tilde{\alpha}_{ij}(x,y)$ instead of the Gaussian weights $\alpha_{ij}$. Note that $\tilde{\alpha}_{ij}(x,y) \le \alpha_{ij}$, where $\tilde{\alpha}_{ij}(x,y)$ are reduced by multiplying with $\beta_{ij}(x,y) \in (0,1]$. The more similar is the spectrum $s(x+i,y+j)$ to the spectrum $s(x,y)$ from the neighbourhood centre, the larger is $\beta_{ij}(x,y)$. Figure 5.8B shows an example of the adaptive weights $\tilde{\alpha}_{ij}(x,y)$ for a neighbourhood containing spectra from two different histological regions; spectra in these regions differ significantly.

In order to eliminate the parameter $\lambda$ in (5.8) which adjusts the adaptivity of $\tilde{\alpha}_{ij}(x,y)$ to the spectra in the neighbourhood of the pixel $(x,y)$, the following approach is proposed. The value of $\lambda(x,y)$ is selected for each neighbourhood separately, in such a way that the largest $\beta_{ij}(x,y)$ in this neighbourhood is 1 and the smallest is $\exp(-2) \approx 0.15$, what leads to

$$\lambda(x,y) = \tfrac{1}{2}\max_{-r\le i,j\le r}\left\{\hat{\delta}_{ij}(x,y)\right\}$$

where $\hat{\delta}_{ij}(x,y) = \delta_{ij}(x,y) - \min_{-r\le i,j\le r}\{\delta_{ij}(x,y)\}$.

### 5.3.3 Efficient implementation with FastMap

Since the weights $\tilde{\alpha}_{ij}(x,y)$ in the distance (5.9) are determined for each pixel separately, the concatenation like in (5.7) cannot be used to obtain $d_{r,\{\tilde{\alpha}_{ij}\},\lambda}(s_1,s_2) = \|\Phi(s_1) - \Phi(s_2)\|_2$ with a simple transformation $\Phi(s)$. Thus, FastMap (Faloutsos and Lin, 1995) is used to find projections

of spectra into $\mathbb{R}^q$ for a given $q$ so that the pairwise distances are similar to those calculated using (5.9). Note that still it is not necessary to calculate all $(n^2-n)/2$ pairwise distances are, but only $n(2q+1)$ thanks to the FastMap trick. Finally, the points found by FastMap are clustered with $K$-means. Furthermore, the mapping into the feature space has negative impact on the runtime of $K$-means clustering, even in the case of Gaussian weights. Therefore, the use of FastMap is advised in any case, since it drastically reduces computation time.

### 5.3.4 Application of spatially aware segmentation

The results presented in this chapter were published by Alexandrov and Kobarg (2011) and Kobarg and Alexandrov (2013). The datasets described in Section 1.2.3 were used. For the rat brain, peak picking procedure with alignment found $p = 71$ peaks and for the tumor data, $p = 62$.

#### Rat brain dataset

Each of the proposed segmentation methods, SA (spatially-adaptive, with Gaussian weights used) and SASA (spatially-adaptive, with structure-adaptive weights), has only three parameters: the pixel neighbourhood radius $r$, the dimension $q$ of the space where FastMap projects the mapped data into, and the number of clusters $K$.

Segmentation maps were produced for $r = 2,3,4$. The FastMap dimension is $q = 20$. The number of clusters is $K = 10$, what by Alexandrov et al. (2010) was found to be representative for this dataset. Figure 5.9 shows an optical image (panel A), the schematic of the anatomical structure (panel B), a segmentation map produced with straightforward clustering of spectra when no spatial relations between spectra are taken into account (panel C), and maps for SA (panels D-F) and SASA methods (panels G-I).

First, one can see that for the segmentation maps produced with both SA and SASA methods reflect the anatomical structure. Some anatomical regions (cortex, hippocampus, corpus callosum and internal capsule, amygdala) are very well represented. Note that the hippocampus has different parts (one in the middle and another close to amygdala) which still have the same colour in the map (mid blue). Some regions are not well represented, e.g. a thin part of thalamus which goes around hypothalamus is not visible. However, as discussed by Alexandrov et al. (2010), this might be not a computational problem but an under-representation of these regions in the processed IMS dataset.

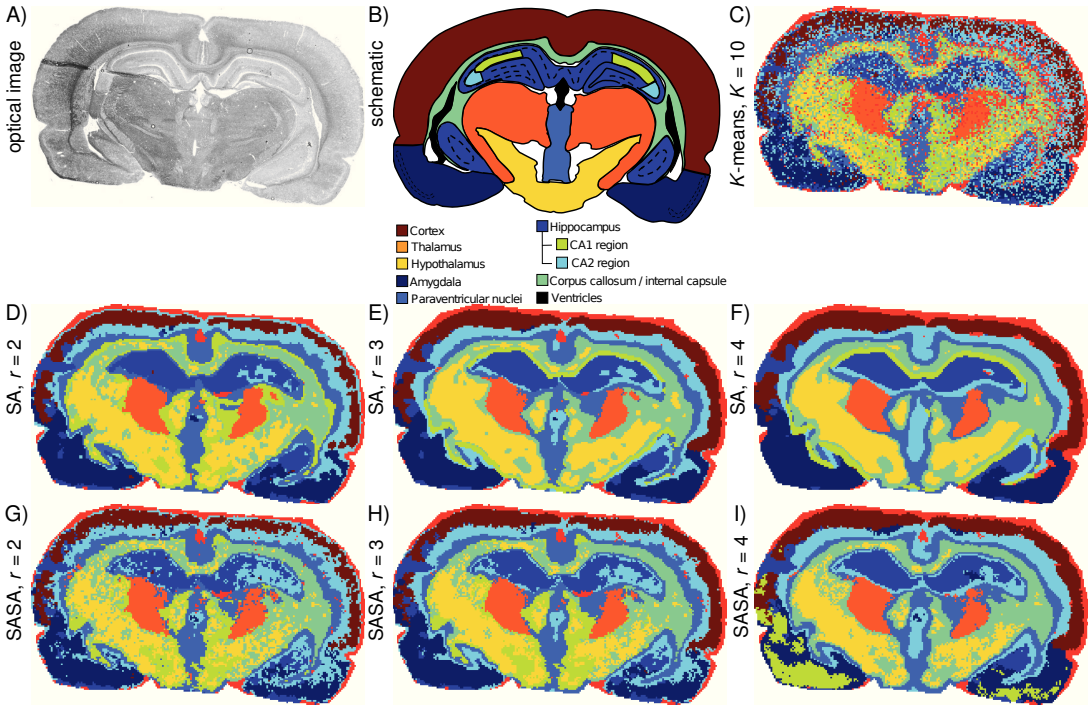



**Figure 5.9: Rat brain dataset.** A. Optical image. B. Schematic based on the rat brain atlas, reproduced from Alexandrov et al. (2010) with permission from the American Chemical Society. C-I. Segmentation maps, $q = 20$, $K = 10$. C. Straightforward $K$-means clustering of spectra. D-F. SA method. G-I. SASA method. (Reproduced from Alexandrov and Kobarg, 2011.)

Second, the proposed methods significantly outperform the straightforward clustering (Figure 5.9C) where strong noise hides details and the whole anatomical regions. For example, in Figure 5.9C amygdala are not separated from hippocampus; hippocampus from the inner part of cortex and from paraventricular nuclei. Importantly, the noise in the segmentation map is a technological and computational artefact but not a property of the brain tissue.

Thus, the conclusion is that the overall quality of the produced segmentation maps for the rat brain dataset is good. Note the blue small region interrupting the left part of cortex (Figure 5.10, region A). This represents a tissue section preparation defect (visible in the optical image as well) when the thin 10 μm tissue section was folded during transferring it onto a glass slide.

The efficiency of the segmentation method was the ultimate goal because existing advanced segmentation methods run several tens of minutes for a dataset. Tens of minutes seems acceptable because it is still less than the dataset acquisition time (several hours). However, this does

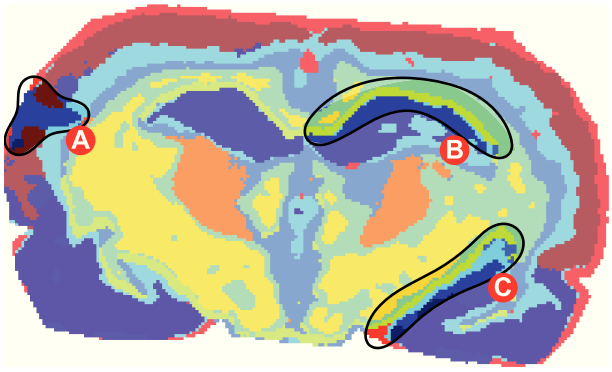



**Figure 5.10: Segmentation map (SA method) for the rat brain dataset.** The parameters are $r = 3$, and $q = 20$, $K = 10$. Region A shows a tissue section preparation defect. Regions B and C highlight an SA-specific artefact, a layer of chartreuse yellow pixels along hippocampus. (Reproduced from Alexandrov and Kobarg, 2011.)

| Dataset | $(n \times p)$ | Method | Neighbourhood radius $r = 2$ | $r = 3$ | $r = 4$ |
|---|---|---|---|---|---|
| Rat brain | $(20185 \times 71)$ | SA | 17 s | 35 s | 49 s |
| | | SASA | 19 s | 32 s | 49 s |
| | | $K$-means* | | 10 s | |
| | | denoising+HDDC* | | 17 min | |
| | | denoising+$K$-means* | | 2 min | |
| | | PCA+hierarchical* | | 25 s | |
| Tumor | $(27960 \times 62)$ | SA | 23 s | 41 s | 62 s |
| | | SASA | 25 s | 39 s | 62 s |

**Table 5.2: Runtimes for producing a segmentation map.** Runtimes on ThinkPad laptop with Intel i5 Core 2.4 GHz; data pre-processing and peak picking are not included; $q = 20$, $K = 10$. *: no neighbourhood is exploited.

not allow one to use segmentation interactively, what is of high importance in imaging applications. Moreover, at the present moment datasets with higher lateral resolution of 20 µm are becoming to be measured (Lagarrigue et al., 2011). If the rat brain section would be measured with 20 µm resolution (instead of 80 µm available for this dataset), this would result into 320,000 spectra. Naturally, such a dataset would demand efficient algorithms.

Table 5.2 shows the runtimes for producing a segmentation map for the considered datasets not including the data loading, preprocessing and peak picking. Incredibly, the proposed methods are almost as efficient as straightforward clustering. The reasons for such efficiency are

| Substep | Rat brain | | Cancer | |
|---|---|---|---|---|
| | SA | SASA | SA | SASA |
| Scaling | 0.01 s | 0.01 s | 0.01 s | 0.01 s |
| Weights | 4 s | 4 s | 6 s | 6 s |
| FastMap | 25 s | 25 s | 32 s | 31 s |
| $K$-means | 5 s | 2 s | 3 s | 3 s |

**Table 5.3: Detailed runtimes for SA and SASA methods.** ThinkPad laptop with Intel i5 Core 2.4 GHz; one iteration of $K$-means is used; $r = 3$, $q = 20$, $K = 10$.

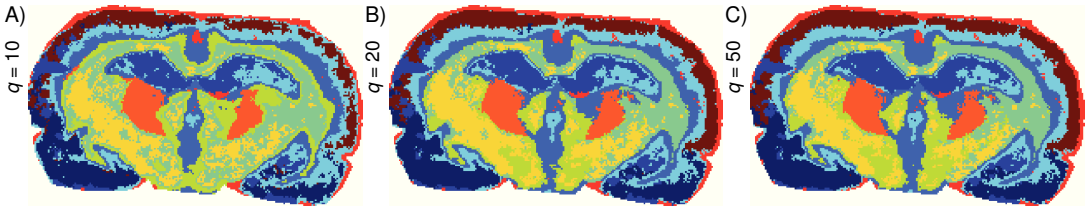



**Figure 5.11: Impact of the FastMap dimension on the segmentation map.** SASA method, $r = 3$, $K = 10$. A. $q = 10$. B. $q = 20$. C. $q = 50$. (Reproduced from Alexandrov and Kobarg, 2011.)

as follows: (1) the proposed approaches for incorporating spatial relations between spectra are computationally simple, (2) the FastMap algorithm has linear complexity $O(nq)$ in the number of spectra and requires just $n(2q+1)$ distances calculated on-the-fly. The latter is especially important if such memory inefficient programming languages as MATLAB are used for implementation.

As for the memory space, the methods are memory-optimized. Both methods need only $n(2r+1)^2$ memory elements for the neighbour indices, $3n$ distance elements in each FastMap iteration and $nq$ memory elements for storing the FastMap projections. The SASA method stores additionally $n(2r+1)^2$ adaptive weights, which at the time of clustering are not needed any longer.

Figure 5.9D-I shows that for both SA and SASA methods, the increase of the neighbourhood radius $r$ makes the maps smoother. This is natural, because for a pixel, more pixels around it are taken into account when calculating (5.6) or (5.8) what helps to reduce the pixel-to-pixel variability. On the other side, small details can be smoothed out, especially by using SA method with non-adaptive weights. For example, the central part of hippocampus in the right half loses its details visible in Figure 5.9D-E but not in Figure 5.9F, as well as the red dot in central part of the cortex (not attributed).

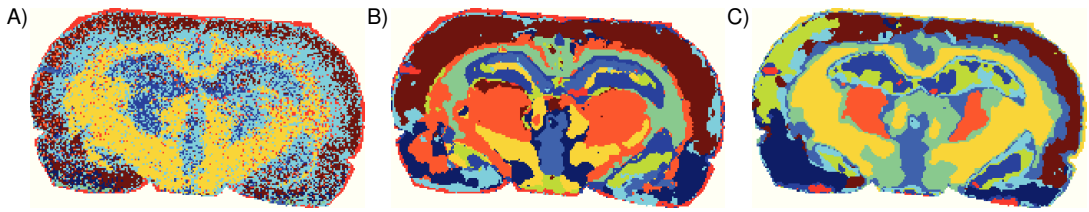



**Figure 5.12: Segmentation maps of compared methods.** A. Hierarchical clustering (Euclidean distance, complete linkage) after PCA-reduction of spectra to 70 % explained variance. B-C. With prior-to-clustering edge-preserving image denoising; (k=10), moderate denoising. B. High Dimensional Discriminant Clustering, reproduced from Alexandrov et al. (2010) with permission from the American Chemical Society. C. *K*-means. (Reproduced from Alexandrov and Kobarg, 2011.)

Recall that the SASA method was constructed so that in a pixel neighbourhood, the weights assigned to the pixels are adaptive; the more different is a spectrum from the spectrum of the central pixel, the less is the weight. Thus, the *averaging* in (5.8) is done mostly among pixels similar to the central one. Comparing the maps for the SA and SASA method in Figure 5.9, one can see that making weights structure-adaptive prevents smoothing out the details and deteriorating of edges between different spatial regions. On the other hand, the SASA maps look noisier.

Comparison of the SA- and SASA-segmentation maps reveals an artefact produced by the SA method. It is a layer of chartreuse yellow pixels along hippocampus which are not visible in the SASA maps and cannot be attributed to any anatomical region. Figure 5.10, regions B and C, highlights the areas of the artefact. It is believed that this is an *averaging* artefact due to weights $\alpha_{ij}$ in equation (5.6) are not adaptive to the data. This is confirmed by the absence of this layer in the SA map with the smallest pixel neighbourhood radius $r = 2$ and in the SASA maps.

The parameter $q$, the dimension of the space FastMap projects the spectra into, is the most tricky among three parameters of the SA and SASA methods. Naturally, increase of $q$ makes the problem high-dimensional and, thus, prone to the curse of dimensionality issue. On the other hand, for any distance-preserving algorithm the quality of projection reduces with decrease of the dimension $q$. As a rule of thumb, $q$ should not be selected greater than $p$ as FastMap would project from $\mathbb{R}^p$ into $\mathbb{R}^q$. For the SA method, this is motivated by the assumption that the mapping (5.7) into $p(2r+1)^2$-dimensional space introduces much redundancy.

Figure 5.11 shows the segmentation maps for the SASA method with $r = 3$ and $K = 10$, for different values of the FastMap dimension $q = 10$, 20, 50. The values of $q$ were selected to be smaller than $p = 71$. One can see that the maps for $q = 20$ and 50 are very similar. The map

for $q = 10$ looks noisier with a possibly artefact region (chartreuse yellow) around the corpus callosum. Possibly, the dimension $q = 10$ is not enough to achieve sufficient quality of the projection in contrast to $q = 20$.

It was already shown that the segmentation maps excel the maps produced with straightforward clustering of spectra (Figure 5.9C), mostly due to reduction of the pixel-to-pixel variability. In comparison consider the method proposed by Deininger et al. (2008), where hierarchical clustering was applied to low-dimensional features extracted with PCA of spectra. Figure 5.12A shows the segmentation map produced with this method. As recommended by Deininger et al. (2008), seven PCA components explaining 70 % variance and the Euclidean distance between extracted features were used. For the Ward linkage need much memory (for this dataset 8 GB was not enough), the complete linkage was exploited. One can see that the segmentation map is very noisy, possibly because the pixel-to-pixel variation in spectra. The average linkage was tested as well; it produces a similarly noisy map.

Next, the methods are compared with advanced segmentation proposed by Alexandrov et al. (2010), where prior-to-clustering edge-preserving denoising of $m/z$ images was used to reduce the pixel-to-pixel variability. The edge-preserving denoising due to Grasmair (2009) requires about 2 minutes for 71 images of the rat brain dataset. The segmentation map from Alexandrov et al. (2010) for $K = 10$ is shown in Figure 5.12. For clustering, High Dimensional Discriminant Clustering (HDDC) was used, the clustering method designed specially for high-dimensional data. However, its main disadvantage is the long runtimes due to using the expectation-maximization algorithm. Moreover, during an M-step it can be unstable when a cluster has a few or no elements, what requires starting it several times with random initializations. Although a new version of HDDC fixing these issues is planned to be included soon in the MIXMOD software (Biernacki et al., 2006), at the time of original publication HDDC is slow. For the rat brain, one iteration takes about 50 seconds. For $K = 10$, at least 20 iterations are necessary because of the mentioned instability, which sums up to approximately 15 minutes.

For this reason, HDDC was replaced with $K$-means in the method proposed by Alexandrov et al. (2010). The resulted segmentation map is shown in Figure 5.12. One can see that the map produced with HDDC is comparable to the maps presented in Figure 5.9. Moreover, as also discussed by Alexandrov et al. (2010), $K$-means seems to be worse than HDDC (artefacts, less regions, detailedness). The SA- and SASA-maps look better than those after $K$-means with prior-to-clustering denoising. The runtimes for methods considered in this section are given in Table 5.2, which are much longer than those for the SA and SASA methods.

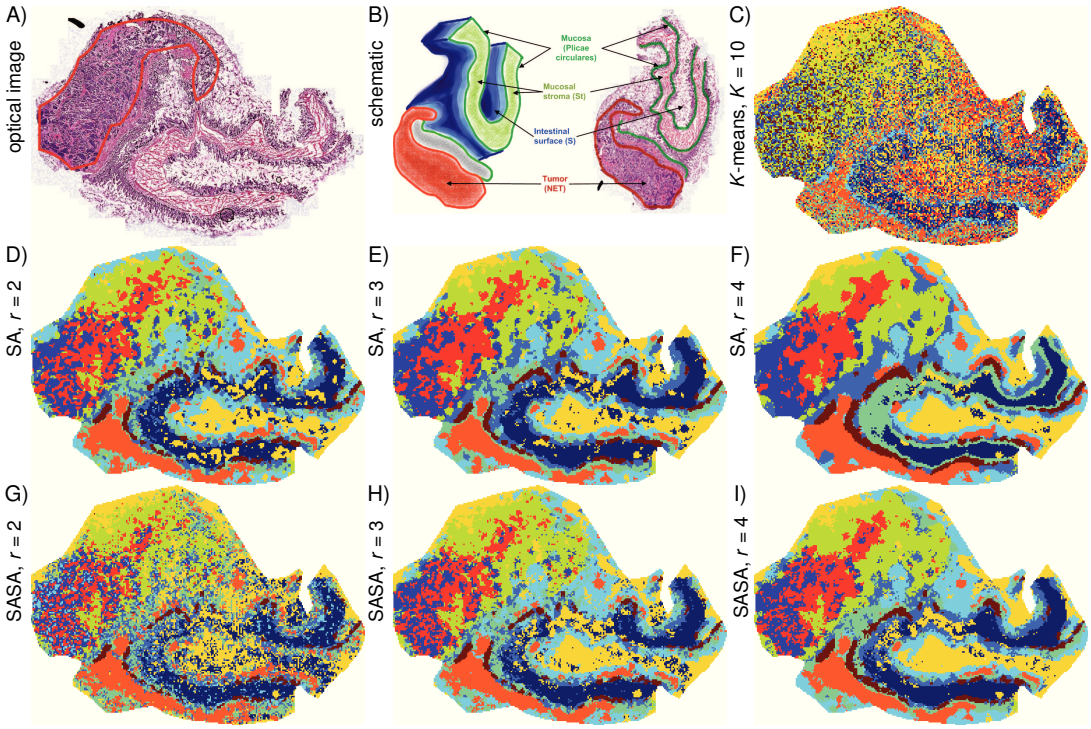



**Figure 5.13: Neuroendocrine tumor dataset.** A. Optical image after H&E staining, the tumor region is selected. B. 3D-structure of the tissue and optical image with main functional structures, (reproduced from Alexandrov et al., 2010, with permission from the American Chemical Society). C-I. Segmentation maps, $K = 10$. C. Straightforward clustering of spectra. D-E. SA method. G-I. SASA method. (Reproduced from Alexandrov and Kobarg, 2011.)

### Neuroendocrine tumor

In this section, the segmentation maps for the second dataset are considered briefly, the neuroendocrine tumor invading the small intestine (ileum). This dataset differs from the rat brain dataset in the following respects: (1) it represents pathology (tumor), (2) the tissue is more complicated, the difference between anatomical regions is not that clear, (3) the tumor area is a heterogeneous composition of tumor cells, tumor stroma, and connective tissue. All this poses a complex challenge for a segmentation algorithm.

Figure 5.13 shows the optical image after H&E staining together with 3D-structure of the tissue and optical image with main functional structures as well as the segmentation maps. First, the tumor region is separated from the rest and is represented in three colours: blue, red, and

chartreuse yellow. This corresponds to results shown by Alexandrov et al. (2010), although there the blue and red regions have not been separated. Moreover, the anatomical structure is represented, although the tissue flattened when put on the slide. Note the layer of brown pixels which is visible in the map after straightforward clustering (in light blue), which was not found by Alexandrov et al. (2010).

Thus, concluding from the presented results, the segmentation maps for this complex and heterogeneous tissue discriminate the tumor area, highlight the anatomical structure even after the transformation of the tissue, and excel the maps produced with the advanced prior-to-clustering denoising method by Alexandrov et al. (2010).

### Simulated 3D datacube

So far, the proposed method was applied to 2D real-life IMS data. Here, the performance of the method is demonstrated with a simulated 3D IMS dataset (Kobarg and Alexandrov, 2013).

The simulated data was treated like any raw IMS data in the way that standard preprocessing routines were applied before segmentation was computed. Total ion count normalization was applied to the data – such that each intensity vector has the same area under its curve – followed by baseline estimation and their subtraction (Alexandrov and Kobarg, 2011). Furthermore, the number of image channels is reduced to those that contain peaks in the mass spectra (Alexandrov et al., 2010). Standard $K$-means was then applied directly, with constant weights and adaptive weights, each with a neighbourhood of width $w = 5$.

The result is shown for two sections in Figure 5.14. The objects cannot be discriminated if the dataset is clustered directly and is rather split within groups. This prevents the upper rectangle class to be detected even after increasing the number of clusters. While in the case of $K = 6$ three structures are well separated from background, the background itself is further divided into three groups, none of them related to the rectangle. The class of stick pixels is lost within the noise for $K = 5$ even though it can be clearly identified with $K = 4$ or $K = 6$.

Once the approach with spatial information embedding is employed, the segmentation map is not affected by noise. In the case of $K = 4$, the isolated stick structures cannot be found by $K$-means, because the number of classes is too restrictive. As soon as $K \geq 5$, the pixels belonging to this class form their own segment. Also for $K \geq 5$ an artefact starts to appear, namely that the pixels which are located next to two different classes are all put into the same group. Bilateral filtering seems to work perfect in the case of $K = 4$ and $z = 6$, as there appear no misclassifications, however $z = 8$ shows small errors on the edge of both rectangle classes.

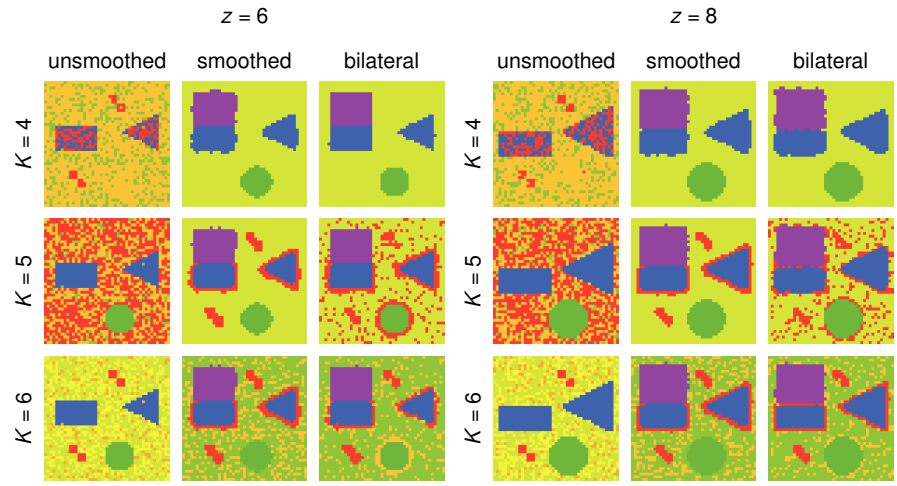



**Figure 5.14: Segmentation map of simulated dataset.** Two spatial slices of the datacube at $z = 6$ and $z = 8$ after $K$-means clustering, without smoothing vs. smoothing ($w = 5$). (Reproduced from Kobarg and Alexandrov, 2013.)

| classes | unsmoothed | smoothed | bilateral |
|---|---|---|---|
| $k = 4$ | 72.32 | 98.72 | 99.38 |
| $k = 5$ | 56.71 | 96.30 | 89.30 |
| $k = 6$ | 67.40 | 75.52 | 76.24 |

**Table 5.4: Clustering performance of simulated dataset.** Balanced accuracy for clustering results against ground truth (in percent).

Furthermore, for $K = 5$ classes the weights are too adaptive to the data and the segmentation map is again affected by noise and an even stronger boundary effect.

As the true classes are known, the comparison of the clustering results is possible with standard classification measures based on confusion matrices (see Section 2.5.4). As each class has different number of members the balanced accuracy is used, which is the mean of sensitivity and specificity. As can be seen in Table 5.4, both types of smoothing outperform direct clustering. The visual deficits of bilateral filtering with $K = 5$ are also visible in the score. Even with $K = 6$, i.e. more classes than truly exist, they outperform direct clustering.

During processing the runtimes of the individual steps were recorded, where data loading and preprocessing are excluded, as those steps are not affected by the algorithm. The most computational effort lies in calculating the lower dimensional representation in $\mathbb{R}^q$ of the data in

the feature space. This step needs approximately two minutes. Clustering of the low dimensional data is carried out in under two seconds. If the reduction with FastMap was not be employed, $K$-means would have to find the clustering of data with the initial $pw^3$ dimensional data. In this setting the algorithm does not finish in under 15 minutes. However, application of the proposed methods was aimed to prove that the algorithm (Alexandrov and Kobarg, 2011) can also be applied to IMS data with three spatial dimensions. Therefore, none of the algorithms was optimized towards speed, but simply adapted for the third spatial dimension.

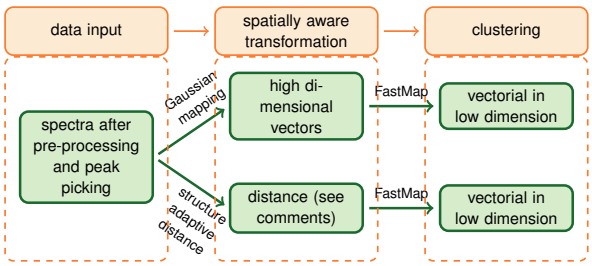
### 5.3.5 Discussion

$K$-means was selected as a clustering algorithm after the FastMap projection because it is a fast and reliable algorithm. Moreover, $K$-means optimizes the Euclidean distances between the points. Thus, performing $K$-means in the space after FastMap projection (for both SA and SASA) is equivalent to minimize the within-point scatter between spectra where the distance between two spectra is calculated using (5.5) or (5.9).

In Alexandrov et al. (2010), denoising of each grey-scale image corresponding to a mass (channel) selected after peak picking is performed. Naturally, a channel-wise processing may be criticized as being prone to lose information presented in a combination of channels. In the methods SA and SASA, channel-wise processing is never done, but the full spectra reduced to a peak list are considered at all times.

As discussed in Results section, the FastMap dimension $q$ is the most tricky parameter. A computational study was performed to investigate the properties of the FastMap projection and allowed to observe that increase of $q$ changes the distances between projections, but only until some value. After this value, the distances between projections stay almost unchanged. Investigating this question can lead to a way of choosing $q$, and, possibly, to the way of finding the intrinsic dimension of a set of points.

The evaluation of produced results is an important problem, especially in an unsupervised framework, where no simple criterion (like total recognition rate) can be computed. The silhouette criterion (Rousseeuw, 1987) of separation between found clusters was tested, but a correspondence between the value of criterion and the visual quality of the maps was not found. Probably, this might be explained by no clear separation between clusters. In the follow up experiment (Kobarg and Alexandrov, 2013) with simulated data, the underlying ground truth could be evaluated and showed high correspondence between the obtained segmentation and the true labels.




| Name of clustering method | Comments |
|---|---|
| Spatially-Aware (SA) | 3 simple parameters, $O(n)$ complexity, needs $O(n)$ memory elements |
| Spatially-Aware Structure-Adaptive (SASA) | 3 simple parameters (the same as in SA), $O(n)$ complexity, needs $O(n)$ memory elements, requires only $n(2q+1)$ distances which are calculated on-the-fly |

**Figure 5.15: Summary of the proposed spatial segmentation methods with comments.**

In general, the evaluation of a spatial segmentation remains an important and unsolved problem in IMS data processing, where no reference or simulated data are provided. Other publications on IMS segmentation, e.g. neither Deininger et al. (2008) nor Alexandrov et al. (2010), do consider this question and present their methods as a data mining tool with extensive support from histologists or biologists estimating the quality of the produced segmentation maps. This is explained by the novelty of this problem and the lack of existing problems and research groups solving this problem. Certainly, in the nearest future the formal evaluation will be necessary, in particular for comparing results of different segmentation. However, any formal evaluation is a complicated task since it requires the ground truth maps for a complex enough dataset. The concept of simulated data as well as reference data will be of great help in this respect.

The algorithm presented in this chapter extends the work by Alexandrov et al. (2010), with the aim to construct an efficient algorithm delivering segmentation maps of at least comparable quality for the datasets from Alexandrov et al. (2010) where comparison is done visually taking into account the anatomical or histological structure of the tissue sample.

The proposed methods are summarized in Figure 5.15, which demonstrates, they can be applied to other IMS modalities, especially those with strong pixel-to-pixel variability (e.g. MALDI, SIMS). Moreover, the SASA method can be of use for processing MALDI-IMS data obtained with the recent Fourier transform ion cyclotron resonance (FTICR) mass analyser (Cornett et al., 2008) which produces spectra much longer than MALDI-TOF-IMS (considered for this work), of order $10^6$. Recall that the memory space and computational complexity of the SASA method does not depend on the dimensionality $p$ of spectra but only on number of spectra $n$ and the FastMap dimension $q$.

The spatial segmentation methods can be applied for segmenting other hyperspectral or multichannel data, for example for terahertz imaging (Brun et al., 2010), or hyperspectral imaging

(Tarabalka et al., 2010) like that one used in German Hyperspectral Satellite Mission (Stuffler et al., 2007). The SASA method is recommended, since for this type of data usually no peak picking is performed and the dimensionality $p$ of data is high. Moreover, in the SASA method, more appropriate distance between spectra can be selected instead of the Euclidean distance in (5.9).

Furthermore, the methods can be used in spatially-3D IMS, where a spectrum is measured for a voxel with spatial coordinates $(x, y, z)$. In this case, the number of spectra increases by another order of magnitude reaching $n = 10^6$. For such $n$, pure distance-based methods which need to keep the full distance matrix in memory become inappropriate in contrast to the here presented methods which are linear in $n$ in memory space.

## 5.4 Conclusion of noise reduction approaches

In this chapter, the focus was entirely on spatial smoothing as a central step of the processing pipeline by Alexandrov et al. (2010). Employing spatial smoothing in the processing pipeline improves the segmentation maps significantly. Spatial smoothing is a standard task in image processing and several ways to perform smoothing can be used. Specifically, from the available smoothing methods, those are of major interest that preserve edges in the images, like the algorithm by Chambolle (2004) does.

However, smoothing takes a considerable amount of time in the processing step. Therefore, an alternative of using as the fast bilateral filter was presented (Tomasi and Manduchi, 1998). Furthermore, since smoothing is a traditional task in image processing, the idea of transferring this task to GPU was examined, since this is hardware specially designed to perform such tasks. Here it was discovered, the originally proposed algorithm by Chambolle (2004) does not parallelize well due to the iterative nature of the algorithm. In contrast, the speed improvement that can be achieved for the bilateral filtering is in the order of two magnitudes. Furthermore, this speed gain is almost independent from the employed window size.

Alternatively, a different concept was presented where the entire information of the spectra together with the spatial neighbours is embedded in a new data matrix. This embedding was first introduced by Alexandrov and Kobarg (2011) from where the results presented in this chapter are taken. The effect of spatial smoothing is controlled by using either Gaussian weights or the bilateral weights.

In general, the inclusion of spatial information is one of the major advantages of IMS in comparison to regular mass spectrometry. With direct tissue analysis several new sources of noise are introduced that makes it more difficult to obtain signal from the data. Due to the number of $m/z$ channels to be processed for a large dataset, not all algorithms known for image smoothing can be applied directly. Furthermore, the decision which algorithm to use should incorporate aspects that the data is stored spectra wise. As such, both the bilateral filter as well as the spatially aware mapping fulfil these considerations and are able to be implemented in a vectorized manner for parallel access to the data. As such they are good candidates to obtain processing advantages on a GPU. From that aspect they achieve the same objective of edge preserving smoothing and obtain similar results. However, the objective is achieved in a more practical way and is as such a superior replacement to the originally proposed spatial smoothing employed by Alexandrov et al. (2010).

# 6 Conclusion and future research

Computational data analysis in imaging mass spectrometry (IMS) is an emerging field. However, due to the complex measurement process and especially due to direct tissue analysis, multiple effects have to be countered (Norris et al., 2007; Schwartz et al., 2003). With the processing pipeline by Alexandrov et al. (2010), which is the overarching topic for this thesis, a couple of computational tasks are covered. From this pipeline, the outcome is a list of relevant $m/z$ values that are further explored in specially targeted follow-up experiments. Since these experiments are time consuming and expensive, improving computational analysis is desired, such that the obtained list of $m/z$ values is reliable. How these follow-up experiments should look has not been established in full detail. Several approaches have been proposed, however are still in the incubation phase of research, rather than forming as established workflows (Rauser et al., 2010a; Veselkov et al., 2014). Multiple issues still need to be investigated until IMS becomes a mature technology that can be used for clinical applications.

This chapter summarizes the work presented in this thesis and contains topics that could not be covered and future goals of research are addressed. Several details have to be considered during the experimental acquisition of IMS data. In Chapter 1 the reader is introduced to the concept of the complexity of IMS data and problems of its processing (Goodwin, 2012). It is obvious, that only specialized computational methods are capable of processing the amount of data that is produced (Bonnel et al., 2011; Hanselmann, 2010; Lee and Gilmore, 2009). The individual steps are highlighted in Chapter 2 with additional mathematical background provided. The brief overview includes different baseline removal algorithms as well as a method of obtaining peaks from the data. Multivariate data analysis with clustering is introduced which is used in IMS for automatic grouping of the data (Chaurand et al., 2004; McCombie et al., 2005). The growing data size makes the use of a hierarchical form of $K$-means necessary which is able to process the data in a quick manner. Hierarchical $K$-means allows the spatial segmentation of large dataset, as shown by Trede et al. (2012b).

Chapter 3 has the focus on providing an even more detailed look at the individual mass spectra, especially on the form of the peak shapes. Furthermore, several concepts of single spectrum simulation (Coombes et al., 2005; House et al., 2011) are expanded and the additional knowledge and special circumstances of whole tissue sections are used to create realistic IMS datasets. This allows the statistical evaluation of algorithms for analysis of IMS data to be performed in an objective and possibly unbiased manner by providing a gold standard dataset.

In Chapter 4 detailed attention is brought to dimensionality reduction methods such as *principal component analysis* (PCA; McCombie et al., 2005) and *non-negative matrix factorization* (NMF; Jones et al., 2011). PCA is one of the most popular tools in the analysis of IMS data (McCombie et al., 2005; Klerk et al., 2007; Trim et al., 2008; Sugiura and Setou, 2010). However, the usage of NMF as a method for soft segmentation offers new interpretability of the data (Jones et al., 2011; Kobarg et al., 2014). NMF can become a tool of spectrum unmixing, when problems like convergence are sufficiently solved (Bartels et al., 2013). Another type of dimensionality reduction is achieved with FastMap (Faloutsos and Lin, 1995) which projects the data into a space with similarities between them preserved.

In Chapter 5, the concept of spatial smoothing is described. MALDI-TOF-IMS data analysis can be improved by reducing noise in the individual images of the data cube. A clustering algorithm without denoising results in a noisy segmentation map, that can be improved if spatial smoothing is applied, especially when it preserves edges between anatomical regions. However, edge-preserving spatial smoothing as proposed by Alexandrov et al. (2010) with the adaptive denoising method by Chambolle (2004) and Grasmair (2009) proved to be a time consuming step (Alexandrov and Kobarg, 2011). Efficient and parallel algorithms are proposed for solving this problem. As demonstrated, the new method produces similar results, while being computationally less expensive. Furthermore, the parallel algorithms can accelerated by using specialized hardware such as graphics processing unit (GPU). The idea of spatial smoothing and efficient data compression is combined to the spatially aware structure adaptive embedding approach. This method proposed by Alexandrov and Kobarg (2011) takes benefit of the bilateral filter weights and the data compression available with FastMap.

Not all encountered questions could be answered in this thesis. Therefore, several more details of the processing pipeline, remaining problems and consecutive evaluation that occur in the context of computational analysis are now listed. In general, the outcome of this processing pipeline is a list of $m/z$ values that correlate to a certain anatomical or histopathological regions and as such are candidates for further exploration (Van de Plas et al., 2007; McDonnell et al.,

2008; Bruand et al., 2011b; Suits et al., 2013). This is especially the case, when the $m/z$ values can discriminate between two different regions. Due to the nature of primarily using data mining tools to obtain this list, no assessment of discrimination quality is given. Neither the pipeline nor the improvements demonstrated later on are capable of doing so. Specialized tools suited for the question of the experiment have to be used. Tools that can be of help are evaluation of discrimination power with *receiver operating characteristic* (ROC) from signal detection theory (Hastie et al., 2009). The discrimination power between two annotated regions is used as a first indicator if a certain peak is a candidate for further investigation. Alternatively, a statistical test can be carried out which requires to formulate a formal hypothesis system (Lagarrigue et al., 2012; Poté et al., 2013; Franceschi et al., 2013).

In all the described methods, the intensity of certain peaks appear in arbitrary units. However, for drug imaging, the actual abundance of the specific peak needs to be known. First solutions are available to access the absolute amount of compound for a given spot and $m/z$ value (Källback et al., 2012; Hamm et al., 2012; Picard et al., 2012). As future work, this is a definite issue that needs to be available in the analysis. Even though not mentioned explicitly, the methods from this thesis do not require modification when quantification is considered. Moreover, it will benefit the entire computational reliance, if quantification standards can be mixed into the tissue during sample preparation.

# Bibliography


Abdelmoula, W. M., Carreira, R. J., Shyti, R., Balluff, B., Zeijl, R. J. M. van, Tolner, E. A., Lelieveldt, B. F. P., Maagdenberg, A. M. J. M. van den, McDonnell, L. A., and Dijkstra, J. (2014): "Automatic registration of mass spectrometry imaging data sets to the Allen brain atlas". *Analytical Chemistry*, 86: 3947–3954.

Aebersold, R. and Mann, M. (2003): "Mass spectrometry-based proteomics". *Nature*, 422: 198–207.

Aerni, H.-R., Cornett, D. S., and Caprioli, R. M. (2006): "Automated Acoustic Matrix Deposition for MALDI Sample Preparation". *Analytical Chemistry*, 78: 827–834.

Alexandrov, T., Meding, S., Trede, D., Kobarg, J. H., Balluff, B., Walch, A., Thiele, H., and Maass, P. (2011): "Super-resolution segmentation of imaging mass spectrometry data: Solving the issue of low lateral resolution". *Journal of Proteomics*, 75: 237–245.

Alexandrov, T. (2012): "MALDI imaging mass spectrometry: statistical data analysis and current computational challenges". *BMC Bioinformatics*, 13: S11.

Alexandrov, T. and Bartels, A. (2013): "Testing for presence of known and unknown molecules in imaging mass spectrometry". *Bioinformatics*, 29: 2335–2342.

Alexandrov, T., Becker, M., Deininger, S.-O., Ernst, G., Wehder, L., Grasmair, M., Eggeling, F. von, Thiele, H., and Maass, P. (2010): "Spatial segmentation of imaging mass spectrometry data with edge-preserving image denoising and clustering". *Journal of Proteome Research*, 9: 6535–6546.

Alexandrov, T., Becker, M., Guntinas-Lichius, O., Ernst, G., and Eggeling, F. (2013): "MALDI-imaging segmentation is a powerful tool for spatial functional proteomic analysis of human larynx carcinoma". *Journal of Cancer Research and Clinical Oncology*, 139: 85–95.

Alexandrov, T. and Kobarg, J. H. (2011): "Efficient spatial segmentation of large imaging mass spectrometry datasets with spatially aware clustering". *Bioinformatics*, 27: i230–i238.

Allen Mouse Brain Atlas (2009): *Allen Institute for Brain Science*. Available from: http://mouse.brain-map.org. Seatle (WA).

Altelaar, A. F. M., Minnen, J. van, Jiménez, C. R., Heeren, R. M. A., and Piersma, S. R. (2005): "Direct molecular imaging of lymnaea stagnalis nervous tissue at subcellular spatial resolution by mass spectrometry". *Analytical Chemistry*, 77: 735–741.

Amstalden van Hove, E. R., Smith, D. F., and Heeren, R. M. (2010): "A concise review of mass spectrometry imaging". *Journal of Chromatography A*, 1217: Mass Spectrometry: Innovation and Application. Part VI, 3946–3954.

Anderson, E. (1936): "The species problem in iris". *Annals of the Missouri Botanical Garden*, 23: 457–509.

Andersson, M., Groseclose, M. R., Deutch, A. Y., and Caprioli, R. M. (2008): "Imaging mass spectrometry of proteins and peptides: 3D volume reconstruction". *Nature Methods*, 5: 101–108.

Andrade, L. and Manolakos, E. S. (2003): "Signal background estimation and baseline correction algorithms for accurate DNA sequencing". *The Journal of VLSI Signal Processing*, 35: 229–243.

Arnold, B., Beaver, R., Groeneveld, R., and Meeker, W. (1993): "The nontruncated marginal of a truncated bivariate normal distribution". *Psychometrika*, 58: 471–488.

Azzalini, A. (1985): "A class of distributions which includes the normal ones". *Scandinavian Journal of Statistics*, 12: 171–178.

Balluff, B. (2013): *MALDI imaging mass spectrometry in clinical proteomics research of gastric cancer tissues*. PhD thesis. Ludwig-Maximilians-Universität München.

Baluya, D. L., Garrett, T. J., and Yost, R. A. (2007): "Automated MALDI matrix deposition method with inkjet printing for imaging mass spectrometry". *Analytical Chemistry*, 79: 6862–6867.

Bartels, A., Dülk, P., Trede, D., Alexandrov, T., and Maaß, P. (2013): "Compressed sensing in imaging mass spectrometry". *Inverse Problems*, 29: 125015.

Behrmann, J. (2013): *Blind Source Separation für MALDI-Imaging*. Bachelor thesis. University of Bremen.

Bemis, K. D., Eberlin, L. S., Ferreira, C., Ven, S. van de, Mallick, P., Cooks, R. G., Stolowitz, M., and Vitek, O. (2013): *Discovering spatio-chemical structure in tissue: CARDINAL Software and methods for analysis of mass spectrometry images*. Poster. American Indian Science and Engineering Society (AISES) National Conference, Denver, CO.

Bemis, K. D., Eberlin, L., Ferreira, C., Cooks, R. G., and Vitek, O. (2012): "Spatial segmentation and feature selection for DESI imaging mass spectrometry data with spatially-aware sparse clustering". *BMC Bioinformatics*, 13: A8.

Ben-Hur, A., Elisseeff, A., and Guyon, I. (2001): "A stability based method for discovering structure in clustered data". In: *Pacific Symposium on Biocomputing 2002: Kauai, Hawaii, 3-7 January 2002*. World Scientific Pub Co Inc, p. 6.

Benninghoven, A. and Loebach, E. (1971): "Tandem mass spectrometer for secondary ion studies". *Review of Scientific Instruments*, 42: 49–52.

Bielow, C., Aiche, S., Andreotti, S., and Reinert, K. (2011): "MSSimulator: Simulation of mass spectrometry data". *Journal of Proteome Research*, 10: 2922–2929.

Biernacki, C., Celeux, G., Govaert, G., and Langrognet, F. (2006): "Model-based cluster and discriminant analysis with the MIXMOD software". *Computational Statistics and Data Analysis*, 51: 587–600.

Bonnel, D., Longuespee, R., Franck, J., Roudbaraki, M., Gosset, P., Day, R., Salzet, M., and Fournier, I. (2011): "Multivariate analyses for biomarkers hunting and validation through on-tissue bottom-up or in-source decay in MALDI-MSI: application to prostate cancer". *Analytical and Bioanalytical Chemistry*, 401: 149–165.

Bouveyron, C., Girard, S., and Schmid, C. (2007): "High-dimensional data clustering". *Computational Statistics & Data Analysis*, 52: 502–519.

Bredies, K. and Lorenz, D. (2011): *Mathematische Bildverarbeitung: Einführung in Grundlagen und moderne Theorie*. Vieweg+Teubner Verlag.

Brignole-Baudouin, F., Desbenoit, N., Hamm, G., Liang, H., Both, J.-P., Brunelle, A., Fournier, I., Guerineau, V., Legouffe, R., Stauber, J., Touboul, D., Wisztorski, M., Salzet, M., Laprevote, O., and Baudouin, C. (2012): "A new safety concern for glaucoma treatment demonstrated by mass spectrometry imaging of benzalkonium chloride distribution in the eye, an experimental study in rabbits". *PLoS ONE*, 7: e50180.

Broersen, A. (2009): *Feature Visualization in Large Scale Imaging Mass Spectrometry Data*. PhD thesis. Technische Universiteit Eindhoven.

Brown, J., Murray, P., Claude, E., and Kenny, D. (2010): "20 µm resolution MALDI imaging of lipid distribution in tissue using lasers operating between 1 kHz and 10 kHz". In: *58th ASMS Conference on Mass Spectrometry*. Salt Lake City, Utah.

Bruand, J., Alexandrov, T., Sistla, S., Wisztorski, M., Meriaux, C., Becker, M., Salzet, M., Fournier, I., Macagno, E., and Bafna, V. (2011a): "AMASS: Algorithm for MSI analysis by semi-supervised segmentation". *Journal Proteome Research*, 10: 4734–4743.

Bruand, J., Sistla, S., Mériaux, C., Dorrestein, P. C., Gaasterland, T., Ghassemian, M., Wisztorski, M., Fournier, I., Salzet, M., Macagno, E., and Bafna, V. (2011b): "Automated querying and identification of novel peptides using MALDI mass spectrometric imaging". *Journal of Proteome Research*, 10: 1915–1928.

Brun, M.-A., Formanek, F., Yasuda, A., Sekine, M., Ando, N., and Eishii, Y. (2010): "Terahertz imaging applied to cancer diagnosis". *Physics in Medicine and Biology*, 55: 4615.

Caprioli, R. M., Farmer, T. B., and Gile, J. (1997): "Molecular imaging of biological samples: localization of peptides and proteins using MALDI-TOF MS". *Analytical Chemistry*, 69: 4751–4760.

Casadonte, R. and Caprioli, R. M. (2011): "Proteomic analysis of formalin-fixed paraffin-embedded tissue by MALDI imaging mass spectrometry". *Nature Protocols*, 6: 1695–1709.

Cazares, L. H., Troyer, D., Mendrinos, S., Lance, R. A., Nyalwidhe, J. O., Beydoun, H. A., Clements, M. A., Drake, R. R., and Semmes, O. J. (2009): "Imaging mass spectrometry of a specific fragment of mitogen-activated protein kinase/extracellular signal-regulated kinase kinase kinase 2 discriminates cancer from uninvolved prostate tissue". *Clinical Cancer Research*, 15: 5541–5551.

Chambolle, A. (2004): "An algorithm for total variation minimization and applications". *Journal of Mathematical Imaging and Vision*, 20: 89–97.

Chaurand, P., Schwartz, S. A., and Caprioli, R. M. (2004): "Assessing protein patterns in disease using imaging mass spectrometry". *Journal of Proteome Research*, 3: 245–252.

Chen, J., Paris, S., and Durand, F. (2007): "Real-time edge-aware image processing with the bilateral grid". *ACM Transactions on Graphics*, 26: 103.

Chiang, M. M.-T. and Mirkin, B. (2010): "Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads". *Journal of Classification*, 27: 3–40.

Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S.-i. (2009): *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. John Wiley & Sons.

Colinge, J. and Bennett, K. L. (2007): "Introduction to computational proteomics". *PLoS Computational Biology*, 3:

Coombes, K. R., Fritsche, H. A., Clarke, C., Chen, J.-N., Baggerly, K. A., Morris, J. S., Xiao, L.-C., Hung, M.-C., and Kuerer, H. M. (2003): "Quality control and peak finding for proteomics data collected from nipple aspirate fluid by surface-enhanced laser desorption and ionization". *Clinical Chemistry*, 49: 1615–1623.

Coombes, K. R., Koomen, J. M., Baggerly, K. A., Morris, J. S., and Kobayashi, R. (2005): "Understanding the characteristics of mass spectrometry data through the use of simulation". *Cancer Informatics*, 1: 41–52.

Cornett, D. S., Frappier, S. L., and Caprioli, R. M. (2008): "MALDI-FTICR imaging mass spectrometry of drugs and metabolites in tissue". *Analytical Chemistry*, 80: 5648–5653.

Crecelius, A. C., Cornett, D. S., Caprioli, R. M., Williams, B., Dawant, B. M., and Bodenheimer, B. (2005): "Three-dimensional visualization of protein expression in mouse brain structures using imaging mass spectrometry". *Journal of the American Society for Mass Spectrometry*, 16: 1093–1099.

Crecelius, A. C., Steinacker, R., Meier, A., Alexandrov, T., Vitz, J., and Schubert, U. S. (2012): "Application of matrix-assisted laser desorption/ionization mass spectrometric imaging for photolithographic structuring". *Analytical Chemistry*, 84: 6921–6925.

Daubechies, I., Defrise, M., and De Mol, C. (2004): "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". *Communications on Pure and Applied Mathematics*, 57: 1413–1457.

Deininger, S.-O., Cornett, D., Paape, R., Becker, M., Pineau, C., Rauser, S., Walch, A., and Wolski, E. (2011): "Normalization in MALDI-TOF imaging datasets of proteins: practical considerations". *Analytical and Bioanalytical Chemistry*, 401: 167–181.

Deininger, S.-O., Ebert, M. P., Fütterer, A., Gerhard, M., and Röcken, C. (2008): "MALDI imaging combined with hierarchical clustering as a new tool for the interpretation of complex human cancers". *Journal of Proteome Research*, 7: 5230–5236.

Deininger, S.-O., Meye, K., and Walch, A. (2012): *Concise Interpretation of MALDI Imaging Data by Probabilistic Latent Semantic Analysis (pLSA)*. Application Note. Bruker Daltonik, Bremen, Germany.

Denis, L., Lorenz, D. A., and Trede, D. (2009): "Greedy solution of ill-posed problems: error bounds and exact inversion". *Inverse Problems*, 25: 115017 (24pp).

Di Marco, V. B. and Bombi, G. G. (2001): "Mathematical functions for the representation of chromatographic peaks". *Journal of Chromatography A*, 931: 1–30.

Ding, C., Li, T., and Peng, W. (2006): "Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method". In: *Proceedings of the National Conference on Artificial Intelligence*. Vol. 21. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 342.

Donoho, D. and Stodden, V. (2003): "When does non-negative matrix factorization give a correct decomposition into parts?" In: *Proceedings of Neural Information Processing Systems*.

Duran, J., Coll, B., and Sbert, C. (2013): "Chambolle's projection algorithm for total variation denoising". *Image Processing On Line*, 3: 311–331.

Ernst, G., Guntinas–Lichius, O., Hauberg–Lotte, L., Trede, D., Becker, M., Alexandrov, T., and Eggeling, F. von (2014): "Histomolecular interpretation of pleomorphic adenomas of the salivary gland by matrix-assisted laser desorption ionization imaging and spatial segmentation". *Head & Neck*, To appear.

Faloutsos, C. and Lin, K.-I. (1995): "FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets". In: *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*. San Jose, California, United States: ACM, pp. 163–174.

Figueiredo, M. A. T. (2005): "Bayesian image segmentation using wavelet-based priors". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.* Vol. 1. San Diego, California, United States, pp. 437–443.

Fisher, R. A. (1936): "The use of multiple measurements in taxonomic problems". *Annals of Eugenics*, 7: 179–188.

Foley, J. P. (1987): "Equations for chromatographic peak modeling and calculation of peak area". *Analytical chemistry*, 59: 1984–1987.

Fonville, J. M., Carter, C. L., Pizarro, L., Steven, R. T., Palmer, A. D., Griffiths, R. L., Lalor, P. F., Lindon, J. C., Nicholson, J. K., Holmes, E., and Bunch, J. (2013): "Hyperspectral visualization of mass spectrometry imaging data". *Analytical Chemistry*, 85: 1415–1423.

Franceschi, P., Giordan, M., and Wehrens, R. (2013): "Multiple comparisons in mass-spectrometry-based -omics technologies". *TrAC Trends in Analytical Chemistry*, 50: 11–21.

Gaussier, E. and Goutte, C. (2005): "Relation between PLSA and NMF and implications". In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 601–602.

Gibb, S. and Strimmer, K. (2011): "Analysis of proteomic data using MALDIquant". In: *Proceedings of the 8th International Workshop on Computational Systems Biology, WCSB 2011*, pp. 49–52.

Glunde, K., Jacobs, M. A., Pathak, A. P., Artemov, D., and Bhujwalla, Z. M. (2009): "Molecular and functional imaging of breast cancer". *NMR in Biomedicine*, 22: 92–103.

Gobom, J., Mueller, M., Egelhofer, V., Theiss, D., Lehrach, H., and Nordhoff, E. (2002): "A calibration method that simplifies and improves accurate determination of peptide molecular masses by MALDI-TOF MS". *Analytical Chemistry*, 74: 3915–3923.

Golbabaee, M., Arberet, S., and Vandergheynst, P. (2010): "Distributed compressed sensing of hyperspectral images via blind source separation". In: *The Asilomar Conference on Signals, Systems, and Computers*. Pacific Grove, CA, USA.

Golbabaee, M., Arberet, S., and Vandergheynst, P. (2013): "Compressive source separation: theory and methods for hyperspectral imaging". *IEEE Transactions on Image Processing*, 22: 5096–5110.

Goodwin, R. J. A. (2012): "Sample preparation for mass spectrometry imaging: Small mistakes can lead to big consequences". *Journal of Proteomics*, 75: 4893–4911.

Goodwin, R. J. A., Pennington, S. R., and Pitt, A. R. (2008): "Protein and peptides in pictures: Imaging with MALDI mass spectrometry". *PROTEOMICS*, 8: 3785–3800.

Grasmair, M. (2009): "Locally adaptive total variation regularization". In: *Scale Space and Variational Methods in Computer Vision*. Ed. by X.-C. Tai, K. Mørken, M. Lysaker, and K.-A. Lie. Vol. 5567. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 331–342.

Guha, S., Rastogi, R., and Shim, K. (2001): "CURE: an efficient clustering algorithm for large databases". *Information Systems*, 26: 35–58.

Guo, D., Gahegan, M., MacEachren, A. M., and Zhou, B. (2005): "Multivariate analysis and geovisualization with an integrated geographic knowledge discovery approach". *Cartography and Geographic Information Science*, 32: 113.

Gustafsson, J. O. R., Eddes, J. S., Meding, S., Koudelka, T., Oehler, M. K., McColl, S. R., and Hoffmann, P. (2012): "Internal calibrants allow high accuracy peptide matching between MALDI imaging MS and LC-MS/MS". *Journal of Proteomics*, 75: 5093–5105.

Gustafsson, J. O. R., Oehler, M. K., McColl, S. R., and Hoffmann, P. (2010): "Citric acid antigen retrieval (CAAR) for tryptic peptide imaging directly on archived formalin-fixed paraffin-embedded tissue". *Journal of Proteome Research*, 9: 4315–4328.

Hamm, G., Bonnel, D., Legouffe, R., Pamelard, F., Delbos, J.-M., Bouzom, F., and Stauber, J. (2012): "Quantitative mass spectrometry imaging of propranolol and olanzapine using tissue extinction calculation as normalization factor". *Journal of Proteomics*, 75: 4952–4961.

Hankin, J. A., Barkley, R. M., and Murphy, R. C. (2007): "Sublimation as a method of matrix application for mass spectrometric imaging". *Journal of the American Society for Mass Spectrometry*, 18: 1646–1652.

Hanselmann, K. M. S. (2010): *Computational methods for the analysis of mass spectrometry images*. PhD thesis. Heidelberg: Ruprecht-Karls-Universität Heidelberg.

Hanselmann, M., Kirchner, M., Renard, B. Y., Amstalden, E. R., Glunde, K., Heeren, R. M. A., and Hamprecht, F. A. (2008): "Concise representation of mass spectrometry images by probabilistic latent semantic analysis". *Analytical Chemistry*, 80: 9649–9658.

Hastie, T., Tibshirani, R., and Friedman, J. (2009): *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.

Heeren, R. M. A. (2014): "Getting the picture: The coming of age of imaging MS". *International Journal of Mass Spectrometry*, in press.

Hiriart-Urruty, J.-B. and Lemaréchal, C. (1996): *Convex analysis and minimization algorithms I*. 2., corrected printing. Vol. 305. Grundlehren der mathematischen Wissenschaften. Berlin: Springer.

House, L. L., Clyde, M. A., and Wolpert, R. L. (2011): "Bayesian nonparametric models for peak identification in MALDI-TOF mass spectroscopy". *Annals of Applied Statistics*, 5: 1488–1511.

Hoyer, P. O. (2004): "Non-negative matrix factorization with sparseness constraints". *Journal of Machine Learning Research*, 5: 1457–1469.

Hubert, L. and Arabie, P. (1985): "Comparing partitions". *Journal of classification*, 2: 193–218.

Hussong, R., Gregorius, B., Tholey, A., and Hildebrandt, A. (2009): "Highly accelerated feature detection in proteomics data sets using modern graphics processing units". *Bioinformatics*, 25: 1937–1943.

Jones, E. A., Deininger, S.-O., Hogendoorn, P. C., Deelder, A. M., and McDonnell, L. A. (2012a): "Imaging mass spectrometry statistical analysis". *Journal of Proteomics*, 75: 4962–4989.

Jones, E. A., Remoortere, A. van, Zeijl, R. J. M. van, Hogendoorn, P. C. W., Bovée, J. V. M. G., Deelder, A. M., and McDonnell, L. A. (2011): "Multiple statistical analysis techniques corroborate intratumor heterogeneity in imaging mass spectrometry datasets of myxofibrosarcoma". *PLoS ONE*, 6: e24913.

Jones, E. A., Zeijl, R. J. M., Andrén, P. E., Deelder, A. M., Wolters, L., and McDonnell, L. A. (2012b): "High Speed Data Processing for Imaging MS-Based Molecular Histology Using Graphical Processing Units". *Journal of the American Society for Mass Spectrometry*, 23: 745–752.

Jungmann, J. H. and Heeren, R. M. A. (2012): "Emerging technologies in mass spectrometry imaging". *Journal of Proteomics*, 75: 5077–5092.

Källback, P., Shariatgorji, M., Nilsson, A., and Andrén, P. E. (2012): "Novel mass spectrometry imaging software assisting labeled normalization and quantitation of drugs and neuropeptides directly in tissue sections". *Journal of Proteomics*, 75: 4941–4951.

Karas, M. and Hillenkamp, F. (1988): "Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons". *Analytical Chemistry*, 60: 2299–2301.

Kaufman, L. and Rousseeuw, P. J. (1990): *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons, Inc.

Kawamoto, T. (2003): “Use of a new adhesive film for the preparation of multi-purpose fresh-frozen sections from hard tissues, whole-animals, insects and plants”. *Archives of Histology and Cytology*, 66: 123–143.

Keenan, M. R. and Kotula, P. G. (2004): “Accounting for Poisson noise in the multivariate analysis of ToF-SIMS spectrum images”. *Surface and Interface Analysis*, 36: 203–212.

Kempka, M., Sjödahl, J., Björk, A., and Roeraade, J. (2004): “Improved method for peak picking in matrix-assisted laser desorption/ionization time-of-flight mass spectrometry”. *Rapid Communications in Mass Spectrometry*, 18: 1208–1212.

Kemsley, E. K. (1996): “Discriminant analysis of high-dimensional data: a comparison of principal components analysis and partial least squares data reduction methods”. *Chemometrics and Intelligent Laboratory Systems*, 33: 47–61.

Klemens, B. (2009): *Modeling with Data: Tools and Techniques for Scientific Computing*. Princton, NJ: Princeton University Press.

Klerk, L. A., Broersen, A., Fletcher, I. W., Liere, R. van, and Heeren, R. M. A. (2007): “Extended data analysis strategies for high resolution imaging MS: New methods to deal with extremely large image hyperspectral datasets”. *International Journal of Mass Spectrometry*, 260: 222–236.

Knochenmuss, R. and Zhigilei, L. V. (2005): “Molecular Dynamics Model of Ultraviolet Matrix-Assisted Laser Desorption/Ionization Including Ionization Processes”. *The Journal of Physical Chemistry B*, 109: 22947–22957.

Kobarg, J. H. and Alexandrov, T. (2013): “Efficient spatial segmentation of hyperspectral 3D volume data”. In: *Algorithms from and for Nature and Life*. Ed. by B. Lausen, D. van den Pol, and A. Ultsch. Studies in Classification, Data Analysis, and Knowledge Organization. Switzerland: Springer International Publishing, pp. 95–103.

Kobarg, J. H. and Maass, P. (2013): *Classification with phase space features*. Internal Report. Project UNLocX: University of Bremen.

Kobarg, J. H., Maass, P., Oetjen, J., Tropp, O., Hirsch, E., Sagiv, C., Golbabaee, M., and Vandergheynst, P. (2014): “Numerical experiments with MALDI Imaging data”. *Advances in Computational Mathematics*, 40: 667–682.

Kobarg, J. H., Tropp, O., Sagiv, C., Rubin, E., and Hirsh, E. (2012): *GPU implementation of BaseLine Algorithms: Efficient transformation of the MALDI data*. Internal Report. Project UNLocX: University of Bremen and SagivTech Ltd.

Kubinyi, H. (1991): “Calculation of isotope distributions in mass spectrometry. A trivial solution for a non-trivial problem”. *Analytica Chimica Acta*, 247: 107–119.

Kwon, D., Vannucci, M., Song, J. J., Jeong, J., and Pfeiffer, R. M. (2008): "A novel wavelet-based thresholding method for the pre-processing of mass spectrometry data that accounts for heterogeneous noise". *Proteomics*, 8: 3019–3029.

Lagarrigue, M., Alexandrov, T., Dieuset, G., Perrin, A., Lavigne, R., Baulac, S., Thiele, H., Martin, B., and Pineau, C. (2012): "New analysis workflow for MALDI imaging mass spectrometry: application to the discovery and identification of potential markers of childhood absence epilepsy". *Journal of Proteome Research*, 11: 5453–5463.

Lagarrigue, M., Becker, M., Lavigne, R., Deininger, S.-O., Walch, A., Aubry, F., Suckau, D., and Pineau, C. (2011): "Revisiting rat spermatogenesis with MALDI imaging at 20 µm resolution". *Molecular & Cellular Proteomics*, 10: M110.005991.

Lan, K. and Jorgenson, J. W. (2001): "A hybrid of exponential and gaussian functions as a simple model of asymmetric chromatographic peaks". *Journal of Chromatography A*, 915: 1–13.

Lance, G. N. and Williams, W. T. (1967): "A general theory of classificatory sorting strategies. 1. Hierarchical systems". *The Computer Journal*, 9: 373–380.

Lange, E. (2008): *Analysis of mass spectrometric data: peak picking and map alignment*. PhD thesis. Freie Universität Berlin.

Lange, E., Gropl, C., Reinert, K., Kohlbacher, O., and Hildebrandt, A. (2006): "High-accuracy peak picking of proteomics data using wavelet techniques". *Pacific Symposium on Biocomputing*, 11: 243–254.

Lee, D. D. and Seung, H. S. (1999): "Learning the parts of objects by non-negative matrix factorization". *Nature*, 401: 788–791.

Lee, D. D. and Seung, H. S. (2001): "Algorithms for non-negative matrix factorization". In: *Advances in Neural Information Processing Systems 13*, pp. 556–562.

Lee, J. L. S. and Gilmore, I. S. (2009): "The application of multivariate data analysis techniques in surface analysis". In: *Surface Analysis – The Principal Techniques*. Ed. by J. C. Vickerman and I. S. Gilmore. 2nd ed. John Wiley & Sons, Ltd. Chap. 10, pp. 563–612.

Lemaire, R., Desmons, A., Tabet, J. C., Day, R., Salzet, M., and Fournier, I. (2007): "Direct analysis and MALDI imaging of formalin-fixed, paraffin-embedded tissue sections". *Journal of Proteome Research*, 6: 1295–1305.

Li, H., Zhang, K., and Jiang, T. (2004): "Minimum entropy clustering and applications to gene expression analysis". In: *Proceedings. 2004 IEEE Computational Systems Bioinformatics Conference, 2004. CSB 2004.* Los Alamitos, CA, USA: IEEE Computer Society, pp. 142–151.

Lin, C.-J. (2007): “On the convergence of multiplicative update algorithms for nonnegative matrix factorization”. *IEEE Transactions on Neural Networks*, 18: 1589–1596.

Liu, Q., Krishnapuram, B., Pratapa, P., Liao, X., Hartemink, A., and Carin, L. (2003): “Identification of differentially expressed proteins using MALDI-TOF mass spectra”. In: *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*. Vol. 2, pp. 1323–1327.

Luts, J., Ojeda, F., Plas, R. V. de, Moor, B. D., Huffel, S. V., and Suykens, J. A. K. (2010): “A tutorial on support vector machine-based methods for classification problems in chemometrics”. *Analytica Chimica Acta*, 665: 129–145.

Maass, P., Kobarg, J. H., and Thiele, H. (2011): *A Phase Space Concept for MALDI Data*. Internal Report. Project UNLocX: University of Bremen.

MacQueen, J. (1967): “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1, pp. 281–297.

Mahoney, M. W. and Drineas, P. (2009): “CUR matrix decompositions for improved data analysis”. *Proceedings of the National Academy of Sciences*, 106: 697–702.

Mantini, D., Petrucci, F., Del Boccio, P., Pieragostino, D., Di Nicola, M., Lugaresi, A., Federici, G., Sacchetta, P., Di Ilio, C., and Urbani, A. (2008): “Independent component analysis for the extraction of reliable protein signal profiles from MALDI-TOF mass spectra”. *Bioinformatics*, 24: 63–70.

McCombie, G., Staab, D., Stoeckli, M., and Knochenmuss, R. (2005): “Spatial and spectral correlations in MALDI mass spectrometry images by clustering and multivariate analysis”. *Analytical Chemistry*, 77: 6118–6124.

McDonnell, L. A., Remoortere, A. van, Velde, N. de, Zeijl, R. J. van, and Deelder, A. M. (2010): “Imaging mass spectrometry data reduction: automated feature identification and extraction”. *Journal of the American Society for Mass Spectrometry*, 21: 1969–1978.

McDonnell, L. A., Remoortere, A. van, Zeijl, R. J. M. van, and Deelder, A. M. (2008): “Mass spectrometry image correlation: quantifying colocalization”. *Journal of Proteome Research*, 7: 3619–3627.

Meding, S. (2012): *Identification of Clinical Markers in Colon Cancer by Tissue Based in situ Proteomics*. PhD thesis. München: Technische Universität München.

Metzger, M. (2012): *Wavelet-Based Baseline Correction for MALDI TOF MS*. Bachelor thesis. University of Bremen.

Mirkin, B. (2005): *Clustering For Data Mining: A Data Recovery Approach*. Vol. 3. Computer Science and Data Analysis. Boca Raton, FL, USA: Chapman & Hall/CRC Press.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003): "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data". *Machine learning*, 52: 91–118.

Morris, J. S., Brown, P. J., Herrick, R. C., Baggerly, K. A., and Coombes, K. R. (2008): "Bayesian analysis of mass spectrometry proteomic data using wavelet-based functional mixed models". *Biometrics*, 64: 479–489.

Morris, J. S., Coombes, K. R., Koomen, J., Baggerly, K. A., and Kobayashi, R. (2005): "Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum". *Bioinformatics*, 21: 1764–1775.

Ng, M. and Huang, J. (2002): "M-FastMap: A modified FastMap algorithm for visual cluster validation in data mining". In: *Advances in Knowledge Discovery and Data Mining*. Vol. 2336. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 224–236.

Ng, R. T. and Han, J. (2002): "CLARANS: A Method for Clustering Objects for Spatial Data Mining". *IEEE Transactions on Knowledge and Data Engineering*, 14: 1003–1016.

Nicolardi, S., Palmblad, M., Dalebout, H., Bladergroen, M., Tollenaar, R. A. E. M., Deelder, A. M., and Burgt, Y. E. M. (2010): "Quality control based on isotopic distributions for high-throughput MALDI-TOF and MALDI-FTICR serum peptide profiling". *Journal of the American Society for Mass Spectrometry*, 21: 1515–1525.

Norris, J. L., Cornett, D. S., Mobley, J. A., Andersson, M., Seeley, E. H., Chaurand, P., and Caprioli, R. M. (2007): "Processing MALDI mass spectra to improve mass spectral direct tissue analysis". *International Journal of Mass Spectrometry*, 260: 212–221.

O'Connor, P. B., Dreisewerd, K., Strupat, K., and Hillenkamp, F. (2013): *MALDI Mass Spectrometry Instrumentation*. In: *MALDI MS*. Wiley-VCH Verlag GmbH & Co. KGaA, pp. 41–104.

Oetjen, J., Aichler, M., Trede, D., Strehlow, J., Berger, J., Heldmann, S., Becker, M., Gottschalk, M., Kobarg, J. H., Wirtz, S., Schiffler, S., Thiele, H., Walch, A., Maass, P., and Alexandrov, T. (2013): "MRI-compatible pipeline for three-dimensional MALDI imaging mass spectrometry using PAXgene fixation". *Journal of Proteomics*, 90: 52–60.

Ostrouchov, G. and Samatova, N. F. (2005): "On FastMap and the convex hull of multivariate data: toward fast and robust dimension reduction". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27: 1340–1343.

Palmer, A. D., Bunch, J., and Styles, I. B. (2013): "Randomized approximation methods for the efficient compression and analysis of hyperspectral data". *Analytical Chemistry*, 85: 5078–5086.

Picard, G., Lebert, D., Louwagie, M., Adrait, A., Huillet, C., Vandenesch, F., Bruley, C., Garin, J., Jaquinod, M., and Brun, V. (2012): "PSAQ™ standards for accurate MS–based quantification of proteins: from the concept to biomedical applications". *Journal of Mass Spectrometry*, 47: 1353–1363.

Piehowski, P. D., Davey, A. M., Kurczy, M. E., Sheets, E. D., Winograd, N., Ewing, A. G., and Heien, M. L. (2009): "Time-of-flight secondary ion mass spectrometry imaging of subcellular lipid heterogeneity: poisson counting and spatial resolution". *Analytical Chemistry*, 81: 5593–5602.

Pock, T., Unger, M., Cremers, D., and Bischof, H. (2008): "Fast and exact solution of Total Variation models on the GPU". In: *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pp. 1–8.

Poté, N., Alexandrov, T., Le Faouder, J., Laouirem, S., Léger, T., Mebarki, M., Belghiti, J., Camadro, J.-M., Bedossa, P., and Paradis, V. (2013): "Imaging mass spectrometry reveals modified forms of histone H4 as new biomarkers of microvascular invasion in hepatocellular carcinomas". *Hepatology*, 58: 983–994.

Prados, J., Kalousis, A., and Hilario, M. (2006): "On preprocessing of SELDI-MS data and its evaluation". In: *19th IEEE International Symposium on Computer-Based Medical Systems, 2006.* Pp. 953–958.

Race, A. M., Steven, R. T., Palmer, A. D., Styles, I. B., and Bunch, J. (2013): "Memory efficient principal component analysis for the dimensionality reduction of large mass spectrometry imaging datasets". *Analytical Chemistry*, 85: 3071–3078.

Rand, W. M. (1971): "Objective criteria for the evaluation of clustering methods". *Journal of the American Statistical Association*, 66: 846–850.

Rauser, S., Deininger, S.-O., Suckau, D., Höfler, H., and Walch, A. (2010a): "Approaching MALDI molecular imaging for clinical proteomic research: current state and fields of application". *Expert Review of Proteomics*, 7: 927–941.

Rauser, S., Marquardt, C., Balluff, B., Deininger, S.-O., Albers, C., Belau, E., Hartmer, R., Suckau, D., Specht, K., Ebert, M. P., Schmitt, M., Aubele, M., Höfler, H., and Walch, A. (2010b): "Classification of HER2 receptor status in breast cancer tissues by MALDI imaging mass spectrometry". *Journal of Proteome Research*, 9: 1854–1863.

Renard, B., Kirchner, M., Steen, H., Steen, J., and Hamprecht, F. (2008): "NITPICK: peak identification for mass spectrometry data". *BMC Bioinformatics*, 9: 355.

Römpp, A. and Spengler, B. (2013): "Mass spectrometry imaging with high resolution in mass and space". *Histochemistry and Cell Biology*, 139: 759–783.

Ross, S. M. (2007): *Introduction to Probability Models*. Elsevier.

Rousseeuw, P. J. (1987): "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". *Journal of Computational and Applied Mathematics*, 20: 53–65.

Rudin, L. I., Osher, S., and Fatemi, E. (1992): "Nonlinear total variation based noise removal algorithms". *Physica D: Nonlinear Phenomena*, 60: 259–268.

Russell, D. H. and Edmondson, R. D. (1997): "High-resolution Mass Spectrometry and Accurate Mass Measurements with Emphasis on the Characterization of Peptides and Proteins by Matrix-assisted Laser Desorption/Ionization Time-of-flight Mass Spectrometry". *Journal of Mass Spectrometry*, 32: 263–276.

Ryan, C. G., Clayton, E., Griffin, W. L., Sie, S. H., and Cousens, D. R. (1988): "SNIP, a statistics-sensitive background treatment for the quantitative analysis of PIXE spectra in geoscience applications". *Nuclear Instruments and Methods in Physics Research Section B*, 34: 396–402.

Sauve, A. C. and Speed, T. P. (2004): "Normalization, baseline correction and alignment of high-throughput mass spectrometry data". In: *Proceedings of the Genomic Signal Processing and Statistics workshop*. Baltimore, MO.

Savitzky, A. and Golay, M. J. E. (1964): "Smoothing and differentiation of data by simplified least squares procedures". *Analytical Chemistry*, 36: 1627–1639.

Schulz-Trieglaff, O., Pfeifer, N., Gropl, C., Kohlbacher, O., and Reinert, K. (2008): "LC-MSsim – a simulation software for liquid chromatography mass spectrometry data". *BMC Bioinformatics*, 9: 423.

Schwartz, S. A., Reyzer, M. L., and Caprioli, R. M. (2003): "Direct tissue analysis using matrix-assisted laser desorption/ionization mass spectrometry: practical aspects of sample preparation". *Journal of Mass Spectrometry*, 38: 699–708.

Schwartz, S. A., Weil, R. J., Johnson, M. D., Toms, S. A., and Caprioli, R. M. (2004): "Protein profiling in brain tumors using mass spectrometry: feasibility of a new technique for the analysis of protein expression". *Clinical Cancer Research*, 10: 981–987.

Sebastiani, F. (2002): "Machine learning in automated text categorization". *ACM Computing Surveys*, 34: 1–47.

Seeley, E. H. and Caprioli, R. M. (2012): "3D imaging by mass spectrometry: a new frontier". *Analytical Chemistry*, 84: 2105–2110.

Shin, H. (2006): *Algorithms for biomarker identification utilizing MALDI TOF mass spectrometry*. PhD thesis. University of Texas at Austin.

Shin, H., Mutlu, M., Koomen, J. M., and Markey, M. K. (2007): "Parametric Power Spectral Density Analysis of Noise from Instrumentation in MALDI TOF Mass Spectrometry". *Cancer Informatics*, 3: 219–230.

Shin, H., Sampat, M. P., Koomen, J. M., and Markey, M. K. (2010): "Wavelet-based adaptive denoising and baseline correction for MALDI TOF MS". *OMICS: A Journal of Integrative Biology*, 14: 283–295.

Solon, E. G., Schweitzer, A., Stoeckli, M., and Prideaux, B. (2010): "Autoradiography, MALDI-MS, and SIMS-MS imaging in pharmaceutical discovery and development". *American Association of Pharmaceutical Scientists Journal*, 12: 11–26.

Spengler, B. (2013): *MALDI-Mass Spectrometry Imaging*. In: *MALDI MS*. Wiley-VCH Verlag GmbH & Co. KGaA, pp. 133–167.

Spraggins, J. and Caprioli, R. (2011): "High-speed MALDI-TOF imaging mass spectrometry: rapid ion image acquisition and considerations for next generation instrumentation". *Journal of The American Society for Mass Spectrometry*, 22: 1022–1031.

Sra, S. and Dhillon, I. S. (2006): *Nonnegative Matrix Approximation: Algorithms and Applications*. Technical Report. Austin, USA: University of Texas at Austin.

Steinbach, M., Karypis, G., and Kumar, V. (2000): *A Comparison of Document Clustering Techniques*. Technical Report. Minneapolis, MN, USA: Department of Computer Science and Engineering, University of Minnesota.

Steinley, D. and Brusco, M. J. (2007): "Initializing k-means batch clustering: A critical evaluation of several techniques". *Journal of Classification*, 24: 99–121.

Stoeckli, M., Chaurand, P., Hallahan, D. E., and Caprioli, R. M. (2001): "Imaging mass spectrometry: A new technology for the analysis of protein expression in mammalian tissues". *Nature Medicine*, 7: 493–496.

Strohalm, M., Strohalm, J., Kaftan, F., Krásný, L., Volný, M., Novák, P., Ulbrich, K., and Havlíček, V. (2011): "Poly[N-(2-hydroxypropyl)methacrylamide]-based tissue-embedding medium compatible with MALDI mass spectrometry imaging experiments". *Analytical Chemistry*, 83: 5458–5462.

Stuffler, T., Kaufmann, C., Hofer, S., Förster, K., Schreier, G., Mueller, A., Eckardt, A., Bach, H., Penné, B., Benz, U., and Haydn, R. (2007): "The EnMAP hyperspectral imager—An advanced optical payload for future applications in Earth observation programmes". *Acta Astronautica*, 61: 115–120.

Sugiura, Y. and Setou, M. (2010): "Statistical procedure for IMS data analysis". In: *Imaging Mass Spectrometry*. Ed. by M. Setou. Springer Japan. Chap. 10, pp. 127–142.

Suits, F., Fehniger, T. E., Végvári, Á., Marko-Varga, G., and Horvatovich, P. (2013): "Correlation queries for mass spectrometry imaging". *Analytical Chemistry*, 85: 4398–4404.

Sun, C. S. and Markey, M. K. (2011): "Recent advances in computational analysis of mass spectrometry for proteomic profiling". *Journal of Mass Spectrometry*, 46: 443–456.

Tanaka, K., Waki, H., Ido, Y., Akita, S., Yoshida, Y., Yoshida, T., and Matsuo, T. (1988): "Protein and polymer analyses up to m/z 100,000 by laser ionization time-of-flight mass spectrometry". *Rapid Communications in Mass Spectrometry*, 2: 151–153.

Tarabalka, Y., Fauvel, M., Chanussot, J., and Benediktsson, J. A. (2010): "SVM- and MRF-based method for accurate classification of hyperspectral images". *IEEE Geoscience and Remote Sensing Letters*, 7: 736–740.

Thiele, H., Heldmann, S., Trede, D., Strehlow, J., Wirtz, S., Dreher, W., Berger, J., Oetjen, J., Kobarg, J. H., Fischer, B., and Maass, P. (2014): "2D and 3D MALDI-imaging: Conceptual strategies for visualization and data mining". *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, 1844: 117–137.

Titulaer, M. K., Siccama, I., Dekker, L. J., Rijswijk, A. L. C. T. van, Heeren, R. M. A., Sillevis Smitt, P. A., and Luider, T. M. (2006): "A database application for pre-processing, storage and comparison of mass spectra derived from patients and controls". *BMC Bioinformatics*, 7: 403.

Tomasi, C. and Manduchi, R. (1998): "Bilateral filtering for gray and color images". In: *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*. Bombay, India: IEEE Computer Society, pp. 839–846.

Trede, D., Kobarg, J. H., Oetjen, J., Thiele, H., Maass, P., and Alexandrov, T. (2012a): "On the importance of mathematical methods for analysis of MALDI-imaging mass spectrometry data". *Journal of Integrative Bioinformatics*, 9: 189.

Trede, D., Schiffler, S., Becker, M., Wirtz, S., Steinhorst, K., Strehlow, J., Aichler, M., Kobarg, J. H., Oetjen, J., Dyatlov, A., Heldmann, S., Walch, A., Thiele, H., Maass, P., and Alexandrov, T. (2012b): "Exploring three-dimensional matrix-assisted laser desorption/ionization imaging mass spectrometry data: Three-dimensional spatial segmentation of mouse kidney". *Analytical Chemistry*, 84: 6079–6087.

Trim, P. J., Atkinson, S. J., Princivalle, A. P., Marshall, P. S., West, A., and Clench, M. R. (2008): "Matrix-assisted laser desorption/ionisation mass spectrometry imaging of lipids in rat brain tissue with integrated unsupervised and supervised multivariant statistical analysis". *Rapid Communications in Mass Spectrometry*, 22: 1503–1509.

Trim, P. J., Djidja, M.-C., Atkinson, S. J., Oakes, K., Cole, L. M., Anderson, D. M. G., Hart, P. J., Francese, S., and Clench, M. R. (2010): "Introduction of a 20 kHz Nd:YVO4 laser into

a hybrid quadrupole time-of-flight mass spectrometer for MALDI-MS imaging". *Analytical and Bioanalytical Chemistry*, 397: 3409–3419.

Upton, G. and Cook, I. (2008): *A Dictionary of Statistics*. Oxford University Press.

Urban, J., Afseth, N. K., and Štys, D. (2014): "Fundamental definitions and confusions in mass spectrometry about mass assignment, centroiding and resolution". *TrAC Trends in Analytical Chemistry*, 53: 126–136.

Van de Plas, R., Pelckmans, K., De Moor, B., and Waelkens, E. (2007): *Spatial querying of imaging mass spectrometry data for the biochemical characterization of anatomical regions in tissue*. Internal Report. Leuven, Belgium: ESAT-SISTA, Katholieke Universiteit Leuven.

Veselkov, K. A., Mirnezami, R., Strittmatter, N., Goldin, R. D., Kinross, J., Speller, A. V. M., Abramov, T., Jones, E. A., Darzi, A., Holmes, E., Nicholson, J. K., and Takats, Z. (2014): "Chemo-informatic strategy for imaging mass spectrometry-based hyperspectral profiling of lipid signatures in colorectal cancer". *Proceedings of the National Academy of Sciences*, 111: 1216–1221.

Vitek, O. (2009): "Getting Started in Computational Mass Spectrometry–Based Proteomics". *PLoS Computational Biology*, 5: e1000366.

Wagner, M., Naik, D., and Pothen, A. (2003): "Protocols for disease classification from mass spectrometry data". *Proteomics*, 3: 1692–1698.

Wagner, M., Graham, D., and Castner, D. (2006): "Simplifying the interpretation of ToF-SIMS spectra and images using careful application of multivariate analysis". *Applied Surface Science*, 252: 6575–6581.

Wang, J. T.-L., Wang, X., Lin, K.-I., Shasha, D., Shapiro, B. A., and Zhang, K. (1999): "Evaluating a class of distance-mapping algorithms for data mining and clustering". In: *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. San Diego, California, United States: ACM, pp. 307–311.

Wang, X., Wang, J. T. L., Lin, K.-I., Shasha, D., Shapiro, B. A., and Zhang, K. (2000): "An index structure for data mining and clustering". *Knowledge and Information Systems*, 2: 161–184.

Watrous, J. D., Alexandrov, T., and Dorrestein, P. C. (2011): "The evolving field of imaging mass spectrometry and its impact on future biological research". *Journal of Mass Spectrometry*, 46: 209–222.

Wehder, L. (2013): *Molekulares Imaging (MALDI-IMS) humaner Kopf-Hals-Tumore und funktionelle Analyse pathologisch exprimierter Proteine am Beispiel von S100A8 und Annexin A5*. PhD thesis. Friedrich-Schiller-Universität Jena.

Williams, B., Cornett, S., Dawant, B., Crecelius, A., Bodenheimer, B., and Caprioli, R. (2005): "An algorithm for baseline correction of MALDI mass spectra". In: *Proceedings of the 43rd annual Southeast regional conference - Volume 1*. ACM-SE 43. Kennesaw, Georgia: ACM, pp. 137–142.

Wolski, W., Lalowski, M., Martus, P., Herwig, R., Giavalisco, P., Gobom, J., Sickmann, A., Lehrach, H., and Reinert, K. (2005): "Transformation and other factors of the peptide mass spectrometry pairwise peak-list comparison process". *BMC Bioinformatics*, 6: 285.

Yang, Y.-L., Xu, Y., Straight, P., and Dorrestein, P. C. (2009): "Translating metabolic exchange with imaging mass spectrometry". *Nature Chemical Biology*, 5: 885–887.

Zhang, T., Ramakrishnan, R., and Livny, M. (1997): "BIRCH: a new data clustering algorithm and its applications". *Data Mining and Knowledge Discovery*, 1: 141–182.

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| BSS | blind source separation |
| DESI | desorption electrospray ionization |
| DHB | 2,5-dihydroxy benzoic acid |
| Da | Dalton |
| EM | expectation-maximization |
| EGH | Exponential-Gaussian hybrid |
| EMG | exponentially modified Gaussian |
| ESI | electrospray ionization |
| FTICR | Fourier transform ion cyclotron resonance |
| GPU | graphics processing unit |
| CHCA | $\alpha$-cyano-4-hydroxycinnamic acid |
| H&E | hematoxylin and eosin stain |
| HDF | hierarchical data format |
| IMS | imaging mass spectrometry |
| ITO | indium-tin-oxide |
| kDa | kilo Dalton |
| MALDI | matrix assisted laser desorption/ionization |
| MDS | multidimensional scaling |
| MRI | magnetic resonance imaging |

| | |
|---|---|
| $m/z$ | mass-to-charge ratio |
| NET | neuroendocrine tumor |
| NMF | non-negative matrix factorization |
| OMP | orthogonal matching pursuit |
| PCA | principal component analysis |
| PLSA | probabilistic latent semantic analysis |
| ROC | receiver operating characteristic |
| SA | 3,5-dimethoxy-4-hydroxycinnamic acid |
| SELDI | surface enhanced laser desorption/ionization |
| SIMS | secondary ion mass spectrometry |
| SVD | singular value decomposition |
| SVM | support vector machine |
| SWT | stationary wavelet transform |
| TIC | total ion count |
| TOF | time-of-flight |