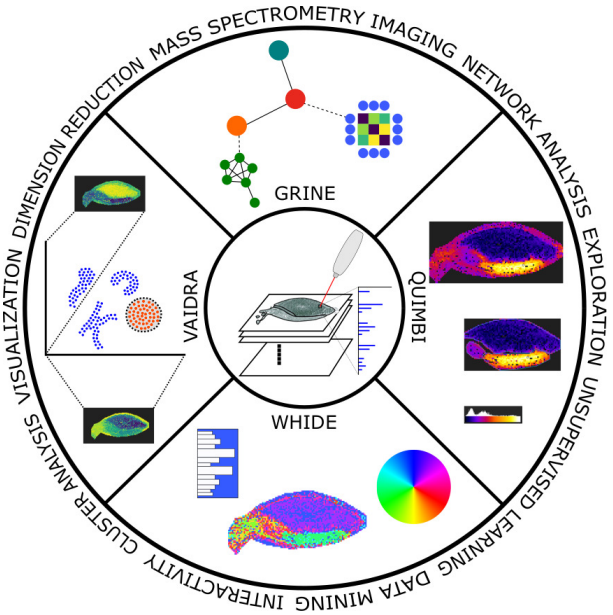
Data-Driven Approaches to Exploratory Visual Analysis of Mass Spectrometry Imaging Data

Karsten Wüllems

November 5, 2020

Data-Driven Approaches to Exploratory Visual Analysis of Mass Spectrometry Imaging Data



submitted by

Karsten Wüllems

for the degree of Dr. rer. nat.

1. Reviewer

Prof. Dr.-Ing. Tim W. Nattkemper

Faculty of Technology

Bielefeld University

2. Reviewer

Prof. Dr. Axel Mosig

Faculty of Biology and Biotechnology

Ruhr-University Bochum

Bielefeld University

Faculty of Technology

Biodata Mining Group

Computational Methods for the Analysis of the Diversity and Dynamics of Genomes

German-Canadian DFG International Research Training Group 1906

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

Der Technischen Fakultät der Universität Bielefeld vorgelegte Dissertation zur Erlangung des akademischen Grades: Doktor der Naturwissenschaften (Dr. rer. nat.)

Dissertation submitted to the Faculty of Technology of Bielefeld University to obtain the academic degree: Doctor rerum naturalium (Dr. rer. nat.)

Karsten Wüllems
Data-Driven Approaches to Exploratory Visual Analysis of Mass Spectrometry Imaging Data
Reviewers: Prof. Dr.-Ing. Tim W. Nattkemper and Prof. Dr. Axel Mosig
Supervisors: Prof. Dr.-Ing. Tim W. Nattkemper and Prof. Dr. Karsten Niehaus

Bielefeld University
Biodata Mining Group
Computational Methods for the Analysis of the Diversity and Dynamics of Genomes
German-Canadian DFG International Research Training Group 1906
Faculty of Technology
Universitätsstraße 25
33615 Bielefeld, Germany

Zusammenfassung

Die bildgebende Massenspektrometrie (MSI) ist eine etablierte und sich stetig weiterentwickelnde Technik zur Analyse der räumlichen Verteilung von kleinen Molekülen in einer Gewebeprobe, in Form von multivariaten Biobildern. Für die Exploration und Analyse von MSI-Daten müssen zwei unterschiedliche Domänen betrachtet werden: die räumliche Domäne, welche Informationen über die räumliche Verteilung von Molekülen und die Morphologie der Probe vermittelt, und die spektrale Domäne, welche Informationen über die Kolokation von Molekülen an spezifischen Positionen in Form von Massenspektren vermittelt. Eine tiefgehende Analyse von MSI-Daten erfordert in der Regel den Einsatz von Expertenwissen, weshalb eine intuitiv bedienbare Software für die visuelle Datenanalyse unerlässlich ist. Daher liegt der Schwerpunkt dieser Arbeit auf computergestützten Ansätzen zur räumlichen, spektralen und räumlich-spektralen Analyse von MSI-Daten.

Um die Kompatibilität mit allen entwickelten visuellen Analysetools zu gewährleisten, habe ich eine Vorverarbeitungspipeline implementiert, welche die üblichen Prozessierungsschritte Alignment, Normalisierung und Peak Picking abdeckt. Zudem ist eine interaktive Methode zur Bestimmung und Subtraktion von Matrix- und Artefaktsignalen aus den Daten, sowie eine Approximation zum Erkennen und Entfernen von Isotopensignalen enthalten.

Um eine schnelle erste Beurteilung der rohen und prozessierten Daten zu ermöglichen, entwickelte ich ein visuelles Analysewerkzeug zur interaktiven Datenexploration auf der Grundlage verschiedener Dimensionsreduktionsergebnisse.

Um die räumiche Domäne der MSI-Daten zu untersuchen, führte ich eine umfassende Studie über die Beziehung zwischen den Daten, der Verwendung von Vorverarbeitungsverfahren, mehreren Ähnlichkeitsfunktionen und verschiedenen Clusteringmethoden durch und schlug einen Workflow vor, um verschiedene Pipeline-Konfigurationen zu bewerten.

Ich entwickelte eine Methode, welche die Beziehung zwischen molekularen Verteilungen auf einer Graphstruktur abbildet, um Community Detection zur Analyse von Clustern kolokalisierter Moleküle anzuwenden. Für die interaktive Analyse entwickelte ich ein zweites visuelles Analysewerkzeug und zeigte dessen Anwendung an realen Daten. Desweiteren habe ich eine Methode entwickelt um interessante Regionen auf der Basis von Clustern von kolokalisierten Molekülen zu approximieren.

Um die spektrale Domäne der MSI-Daten zu untersuchen, schlug ich neue Ansätze zur Berechnung von Segmentierungskarten mit verbessertem Farbkontrast

vor. Für die interaktive Analyse entwickelte ich eine Neuauflage eines visuellen Analysewerkzeuges und zeigte die Anwendung dieser Ansätze auf reale Daten.

Anschließend zeige ich ein Beispiel dafür, wie die räumliche und die spektrale Domäne kombiniert werden können. Zu diesem Zweck entwickelte und evaluierte ich eine weitere Neuauflage eines visuellen Analysewerkzeuges, um die Ähnlichkeit spektraler Kollokation durch Durchsuchen der räumlichen Domäne zu analysieren.

Zusammenfassend trägt diese Arbeit zu dem interdisziplinären Gebiet der molekularen Biowissenschaften, der Informatik und der visuellen Analyse bei, indem sie verschiedene neue Ansätze für die halbautomatische interaktive visuelle Analyse von MSI-Daten liefert.

Abstract

Mass spectrometric imaging (MSI) is an established and still evolving technique for the analysis of the spatial distributions of small molecules in tissue samples in form of multivariate bioimages. For the exploration and analysis of MSI data two different domains have to be considered: the spatial domain, which conveys information about the spatial distribution of molecules and the morphology of the sample, and the spectral domain, which conveys information about the co-location of molecules at a specific position in form of mass spectra. An in-depth analysis of MSI data usually requires the use of expert knowledge, which makes intuitively usable software for visual data analysis essential. Thus, this thesis is focused on computational approaches for the spatial, spectral and spatiospectral visual analysis of MSI data.

To ensure compatibility with all of the visual analysis tools developed, I implemented a (pre-)processing pipeline that covers the usual processing steps of signal alignment, normalization and peak picking. It also includes an interactive method for the determination and subtraction of matrix and artifact signals from the data and an approximation to detect and remove isotope signals.

To enable a fast initial assessment of raw and processed data, I developed a visual analysis tool for interactive data exploration based on various dimensional reduction results.

To investigate the spatial domain of MSI data, I conducted a comprehensive study of the relationships between the data, the use of pre-processing procedures, several similarity functions and different clustering methods, and proposed a workflow to evaluate different pipeline setups.

I developed a method that maps the relationship between molecular distributions on a graph structure to apply community detection to analyze clusters of co-localized molecules. For interactive analysis, I developed a second visual analysis tool and demonstrated its application on real data. Furthermore, I propose a method to approximate regions of interest based on clusters of co-localized molecules.

To investigate the spectral domain of MSI data, I proposed new approaches to compute segmentation maps with enhanced color contrast. For interactive analysis, I developed a refurbished version of a visual analysis tool and demonstrated the application of these approaches to real data.

Finally, I show an example of how the spatial and spectral domain can be combined. For this purpose, I developed and evaluated a refurbished version of another visual

analysis tool to analyze the similarity of spectral co-location by browsing the spatial domain.

In summary, this thesis contributes to the interdisciplinary field of molecular life sciences, computer science and visual analytics by providing various new approaches for the semi-automated interactive visual analysis of MSI data.

Acknowledgment

Done! Finally I am here and write my last sentences. It was a long way, with many obstacles. Most of the time it was difficult to separate professional and personal life, and the glorified work-life balance became a tangled something that shaped my life. Research can be frustrating and it can make you look into the abyss. But it can also be very satisfying and rewarding. What should I say? Those who strive for great things must not fear to fall from time to time. I am proud of everything I have achieved, but this is not only my success, and I think this is the right place to thank all the people who believed in me, even in the moments when I did not.

I would like to thank all DiDies and my colleagues from the Biodata Mining Group for the valuable conversations and discussions we had. I was able to consult you in every situation without any of you complaining once. Many thanks to Roland and Heike, who always tried to make everything run as smoothly as possible, and to Karsten and Hanna, who I could always consult for biological questions. Furthermore, I thank the students who supported me during their projects, especially Daniel, Jonas and Christian. You were so highly motivated in your projects and invested much more effort and passion than I could ever expect from a student.

Thank you, Lukas, who was not only a good friend and roommate throughout my years of study, but also the best “rubber duck” I could wish for. You never complained and were always open to fruitful discussions.

Thank you, Annika, the short time we shared an office was a lot of fun, and I wish I had invited you for collaboration much earlier.

I also thank my family and friends for always supporting me and distracting me when things did not go the way I would have liked. Without you, the hard times would certainly have overwhelmed me.

Special thanks go to my partner in life, Carmen. The person who accompanied me in the best and in the worst times. I assure that she had to endure a lot during my studies, but she did not complain. She supported me in every situation in life, and she was more important for the completion of my studies than she can probably imagine.

Finally, I would like to thank my supervisor and mentor Tim W. Nattkemper. He not only supported me with suggestions, advice and discussions on scientific questions but was also a strong pillar of support during the small mental breakdowns that every student experiences from time to time.

If I have forgotten anyone in this long list of people who deserve my gratitude, I want you to know that I am grateful.

Most Important Contributions and Findings

ProViM – An Instrument Independent Pre-processing Pipeline

With ProViM, we present a new mass spectrometry imaging processing pipeline. The pipeline is designed to work independently of the mass spectrometry imaging instrument used. ProViM provides processing steps for: 1. spectrum alignment and normalization, 2. detection and subtraction of matrix and artifact profile spectra and 3. peak picking and deisotoping. None of these steps require any prior knowledge about the sample, which is especially useful for matrix detection and deisotoping. ProViM is compatible with the vendor-independent imzML format. The results are formatted into the more efficient HDF5 storage format. All steps can be executed fully automated. However, steps two and three also allow manual intervention, which we recommend. The manual intervention gives the user full control over sensitive filtering steps and allows fine-tuning based on expert knowledge [Wüllems et al., in revision].

VAIDRA – Interactive Visual Exploration and Annotation of Dimension Reductions for Mass Spectrometry Imaging Data

The interactive analysis of different dimensional reduction methods can be effectively used to discover morphologically interesting areas based on a similar molecular composition. We have developed a visual analysis tool that allows the exploration of low-dimensional embeddings while preserving the context of the sample morphology. The tool is also equipped with an annotation and export functionality to provide future compatibility with (pre-)processing procedures such as the matrix and artifact detection and subtraction step of ProViM [Wüllems and Nattkemper, 2020].

Quantification of Similarity Between Molecular Images

The spatial features of mass spectrometry imaging data can vary strongly based on the morphology of the biological sample and the measuring instrument. We investigated whether and how these features influence the choice of an effective similarity function to quantify the similarity between molecular images (m/z-images) for subsequent cluster analysis. For this purpose, similarity functions based on a variety of different approaches were investigated. We also investigated whether image processing influences the quantification of similarity. We discovered

that no simple generalization is possible and that any mass spectrometry imaging data set requires an individual evaluation to obtain optimized results.

SoRC – Evaluation and Customization of Pipeline Setups for Spatial Co-Localization Analysis

We developed a workflow scheme for evaluating different pipeline setups, for the quantification of pairwise similarity between molecular images (m/z-images) with subsequent cluster analysis. The proposed workflow scheme uses cluster indices to score a set of given setups against each other. The ranking is then visualized as a combination of table and bar chart [Wüllems and Nattkemper, 2020].

Approximation of Regions of Interest in the Spatial Domain

To propose regions of interest based on molecular co-localization, we apply clustering to molecular images (m/z-images), which results in clusters of co-localized molecules (m/z-values). After that, representatives are computed for each cluster. These representatives are used to approximate and evolve potential regions of interest.

COBI-GRINE – A Community Detection Approach for Spatial Co-Localization Clustering

The contribution of COBI–GRINE consists of two parts: First, we propose a method for projecting the similarity between molecular images (m/z-images) onto a graph structure, which we refer to as mass-to-charge (m/z)-image similarity graph. This projection requires the computation of a pairwise similarity matrix, followed by a transformation into an adjacency matrix, which represents the graph. The graph is then used to compute hierarchical communities. Second, we developed an interactive visual analysis tool to explore an m/z-image similarity graph and to investigate its communities. The tool is complemented by functionalities for user guidance and manual refinement of the communities [Wüllems et al., 2019; Wüllems, 2017.

WHIDE2 – Interactive Segmentation Maps to Explore Similarities in Molecular Composition

We present WHIDE2, a revised version of WHIDE [Kölling et al., 2012]. WHIDE2 is a visual analysis tool for the interactive exploration of segmentation maps. The associated segmentation maps can be computed in two ways: First, by using the hierarchical hyperbolic self-organizing map (H^2SOM) algorithm, or second, by

using an arbitrary clustering method in combination with dimension reduction to project either the entire data or the clustering result into a two-dimensional embedding. We also propose an algorithm to improve the colormapping of H^2SOM-based segmentation maps by adjusting the colors based on the similarities between cluster prototypes.

QUIMBI – Spatial Browsing to Explore Similarities in Molecular Composition in Real-time

We present a revised version of QUIMBI, an interactive visual analysis tool for spatial browsing of similarities in molecular composition in real-time. QUIMBI uses WebGL to parallelize the computation of pairwise similarities between a reference spectrum and all other spectra of a sample. By using the spatial positions associated with each spectrum, the similarity values can be visualized on the sample by pseudo-coloring. The result can be interpreted as a similarity heat map of the molecular compositions. We show that this method is well suited to examine morphological features based on molecular compositions and that similarity relationships can be found that cannot be detected when analyzing single molecular images (m/z-images) [Wüllems et al., in revision].

Contents

1 Introduction

> *Progress is the realisation of utopias.*
>
> **— Oscar Wilde**
> (Author and poet)

Every life on earth is built upon DNA. But DNA alone does not make an organism. Biological life is highly complex and on a molecular level, most processes needed to form a living organism can be categorized into one of the four omics sciences: genomics, transcriptomics, proteomics and metabolomics. On a metaphoric level we can break it down to the analogy of a production line:

Genes → Transcripts → Proteins → Metabolites

Sticking with this metaphor genes relate to lists of ingredients that are used by transcripts to build and regulate machines that relate to proteins. Those machines are used to process goods, which relate to metabolites. This metaphor makes clear, that to keep an organism running, not only each part of this line itself plays an important role, but also their interaction. While the invention of DNA sequencers was a technological breakthrough for research in genomics and transcriptomics, the same can be said for mass spectrometers in the areas of proteomics and metabolomics.

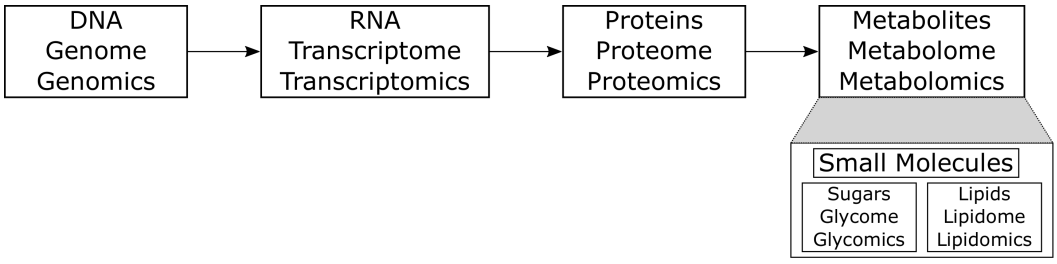


Fig. 1.1.: Omics research fields – Outline of the four main omics research branches. Each container contains the term of the biological entity, the term for their entirety and the term of the omics field. Opposed to the other entities, the term metabolite is a generic term for small molecules within an organism, such as sugars and lipids, which form two large subgroups that are of great interest.

Generally, a mass spectrometer consists of three main components: an ion source, a mass analyzer and a detector (outlined in Figure 1.2). The ion source generates electrically charged molecules by ionization. The ionized molecules (ions) are

transferred to the mass analyzer where they are separated according to their m/z-ratio, also called m/z-values or m/z-channels. This thesis will stay with the term m/z-values. Thereafter, the detector measures either the charged induced or the current produced when ions hit or pass a surface. The result can be represented as a mass spectrum, which represents the ion signal as a function of the m/z-values. This ion signal can be interpreted as the relative abundance of molecules of a specific m/z-value, which is the reason why it is referred to as intensity.

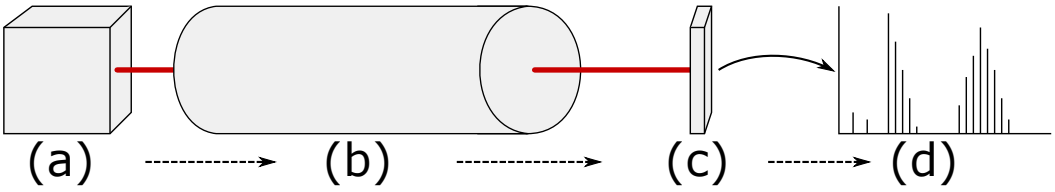


Fig. 1.2.: Basic parts of a mass spectrometer – A mass spectrometer consists of three basic parts: (a) the ionization source, (b) the mass analyzer and (c) the detector. Detected signals can be represented as a mass spectrum (d).

Regardless of their biological scale, whether it is cell, tissue or organ level, many biological pathway activities are highly spatially localized [103, 3, 115]. Therefore, if the ion source cannot spatially resolve its measurements, there is a lot of important information that cannot be captured. To tackle this issue there is a special subtype of mass spectrometers called imaging mass spectrometers.

1.1 Mass Spectrometry Imaging

Mass spectrometry imaging (MSI) is a subtype of mass spectrometry (MS), which measures mass spectra at multiple different positions across a sample [74]. By keeping track of these positions the intensities of specific m/z-values can be mapped and presented as a pseudocolored image. These images are called m/z-images. The result of an MSI measurement is a multivariate data set, which is outlined in Figure 1.3. A two-dimensional MSI measurement results in a data cube consisting of three dimensions, two spatial dimensions and one spectral dimension. This means that the selection of a specific pixel in the spatial plane will result in a mass spectrum, while the selection of a two-dimensional slice at a specific m/z-value will result in an m/z-image. A more detailed definition of the MSI data cube is given in Section 1.1.1. Although methods for three-dimensional MSI measurements exist, this thesis will only cover two-dimensional MSI approaches.

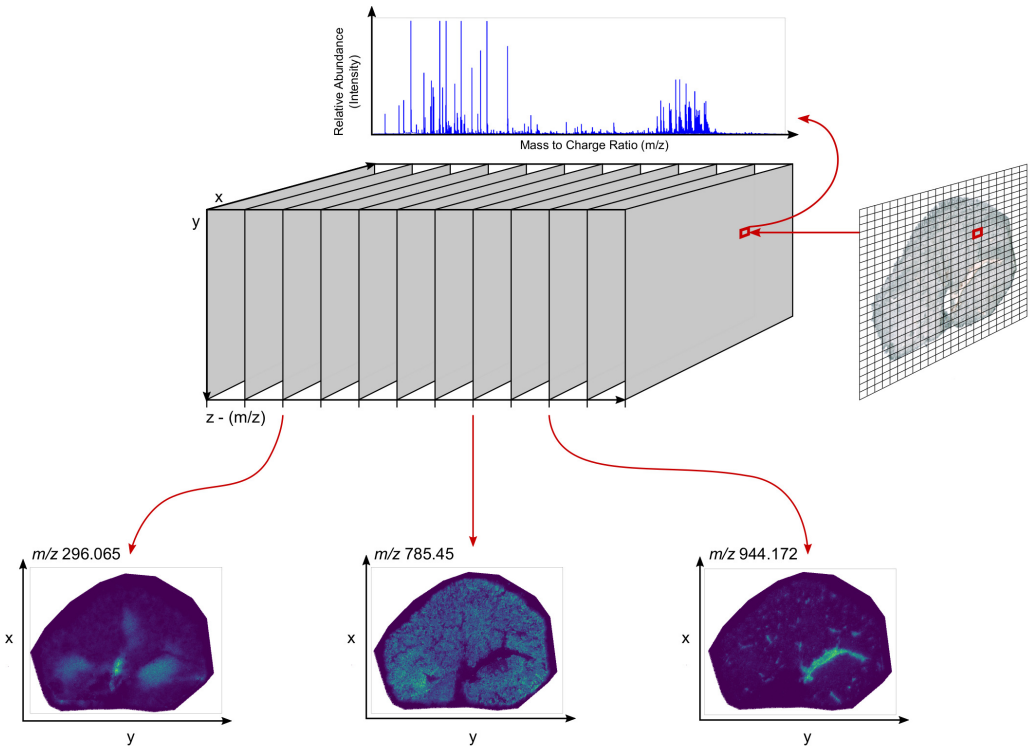


Fig. 1.3.: Mass spectrometry imaging data cube – A mass spectrometry imaging data cube consists of three dimensions. Two spatial dimensions (x, y) and one spectral dimension (z). The selection of a two-dimensional slice at a position z results in an m/z-image, while the selection of a position (x, y) results in a mass spectrum.

These days MSI is still most commonly used in the areas of chemistry, biology and biochemistry. However, there is a continuous increase of application in areas such as medicine, pharmaceutics, pathology and plant research [88].

1.1.1 Formal Definition of a Mass Spectrometry Imaging Data Cube

A two-dimensional MSI data set is defined as a three-dimensional data cube $\mathcal{I} \in \mathbb{R}^{(H \times W \times Z)}$, where $H \times W$ defines the spatial dimension and Z defines the spectral dimension. The spatial dimension consists of $H \in \mathbb{N}^+$ rows and $W \in \mathbb{N}^+$ columns of multivariate pixels $\mathcal{I}_{h,w}$, with $h \in \{0, \ldots, H-1\}$ and $w \in \{0, \ldots, W-1\}$. The set of all multivariate pixel positions is defined as $p = \{(h, w) \mid h \in \{0, \ldots, H-1\}, w \in \{0, \ldots, W-1\}\}$. Although the data set is technically defined as a cube, the measured area does not necessarily have to be rectangular. To take this into account the subset $\rho \subseteq p$ of all multivariate spectral positions describes all positions where the MSI instrument conducted a mass spectrometry measurement, i.e. a mass spectrum has been recorded. It follows, that every multivariate spectral pixel $\mathcal{I}_{h,w}$, with $(h, w) \in \rho$,

corresponds to a mass spectrum. The spectral dimension of every mass spectrum consists of $Z \in \mathbb{N}^+$ measured intensity values, each of which corresponds to a specific m/z-value. For every m/z-value, a two-dimensional slice $\mathcal{I}_z \in \mathbb{R}^{(H \times W)}$, with $z \in \{0, \ldots, Z-1\}$ represents an m/z-image, i.e. the molecular distribution of the z'th m/z-value. The individual pixels $\mathcal{I}_{h,w,z} \in \mathbb{R}$ of a specified m/z-image represent either the intensity value of m/z-value z at position (h, w), if $(h, w) \in \rho$ or $\mathcal{I}_{h,w,z} = 0$, otherwise.

1.1.2 Matrix-Assisted Laser Desorption/Ionization

Matrix-Assisted Laser Desorption/Ionization (MALDI) was not only one of the first ionization techniques used for MSI [18], it is also still the most commonly used [88]. For this technique, the sample has to be embedded in a matrix material. The matrix is required for the ionization process and acts as a mediator to absorb the laser energy and transfers it to the sample. While there is still ongoing research on the exact mechanism, the current state of knowledge is that the absorption of laser energy ionizes the matrix molecules and initiates an ablation process for matrix and sample. The ablation process transfers both into the gas phase. It is assumed that the ionized matrix protonates or deprotonates the sample molecules during this process [52].

From the described mechanism follow three important properties for the generation of data:

1. MALDI is a destructive measurement method in most applications, meaning that each sample can be measured only once.

2. The maximum spatial resolution depends on the diameter of the laser.

3. In most cases, the sample surface and the matrix application are both uneven. This unevenness may affect the amount of ionized molecules, which subsequently affect the detection and the resulting mass spectra.

Alternative technologies for ionization, such as desorption electrospray ionization (DESI) or secondary ion mass spectrometry (SIMS) will not be discussed in this thesis.

1.1.3 Time of Flight

Time of flight (ToF) is a mass analyzer technique, which is often coupled with MALDI. The idea is to accelerate ions by an electrical field and to measure the time they need to pass an electrical field-free region, called drift region, and to reach the detector. Each ions' velocity depends on their weight and charge. The time of flight can be used to calculate the velocity and consequently the m/z-value, hence the name. Instead of using a linear drift region, a reflector can be used. Depending on the mass and charge the ions will dive deeper or shallower into the reflection area and become differently re-accelerated. This results in an increased resolution of the measurement [68, 67]. An outline of this principle is illustrated in Figure 1.4. However, independent of the use of a reflector, ToF is a fast applicable but low-resolution measurement technique, which can result in alignment problems and peak ambiguities.

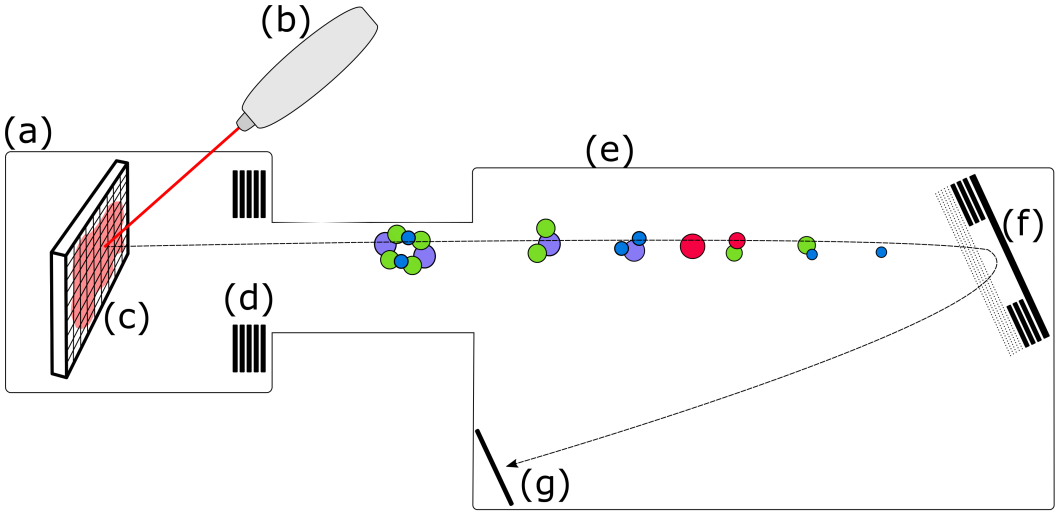


Fig. 1.4.: MALDI-ToF instrument – Outline of a MALDI-ToF instrument setup: (a) ion source (b) laser (c) matrix covered sample, (d) electric acceleration field, (e) drift region, (f) reflector and (g) detector.

1.1.4 Orbitrap

Opposed to ToF, the Orbitrap technology is a measurement technique with high mass resolution [99]. Its principle is outlined in Figure 1.5. The idea is that ions are accelerated by an electric field and injected tangentially into the Orbitrap chamber. This chamber consists of an inner spindle-like electrode and an outer hull-like electrode, which is usually split into two halves. Because the injection is decentralized, the ions will rotate along the central axis of the inner electrode,

resulting in helix-like movements. Different ions will have different rotational patterns, which are measured by detectors in the outer electrode. With the help of the Fourier transform, the respective m/z-values can be calculated from these rotational patterns [66].

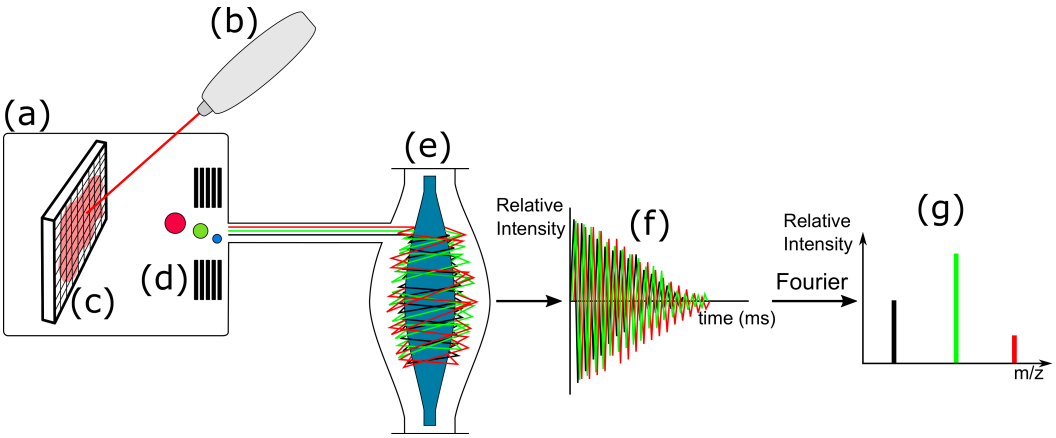


Fig. 1.5.: MALDI-Orbitrap instrument – Outline of a MALDI-Orbitrap instrument setup: (a) ion source (b) laser (c) matrix covered sample, (d) electric acceleration field, (e) Orbitrap (spindle and hull), (f) detected signal and (g) mass spectrum.

1.2 Analysis Approaches in Mass Spectrometry Imaging

An MSI data set features two main domains. Analysis can be performed either along the spatial dimension (spatial domain) or the spectral dimension (spectral domain) [43]. Both approaches have their use cases. However, depending on the domain to be analyzed, data types and result formats will differ. The spatial domain deals with sets of m/z-images, which leads to the field of image analysis. The spectral domain deals with sets of mass spectra, which leads to the field of signal analysis. To differentiate between those two domains and the corresponding analysis approaches, they are addressed by two different terms. The term co-localization is used to describe the degree of similarity between m/z-images. The term molecular composition is used in connection with the description of the degree of similarity between mass spectra.

Co-localization Co-localization describes a function $f(\mathcal{I}_z, \mathcal{I}_{z'})$ that quantifies the relation between the spatial intensity value distribution patterns between two m/z-images $\mathcal{I}_z$ and $\mathcal{I}_{z'}$. In general, a strong co-localization describes a strong spatial

overlap of similar intensity values between two patterns, while the opposite is true for a weak co-localization. If the intensity value distribution patterns of two m/z-images are dissimilar, it can be said that they are not co-localized at all.

Molecular composition comparison The comparison of molecular compositions describes a function $f(\mathcal{I}_{h,w}, \mathcal{I}_{h',w'})$ that measures the relationship between two mass spectra $\mathcal{I}_{h,w}$ and $\mathcal{I}_{h',w'}$. In general, a similar molecular composition describes that two mass spectra share a similar distribution between the measured intensity values at the individual m/z-values. The opposite is true for a dissimilar molecular composition.

1.3 Unsupervised Data-Driven Analysis

Unsupervised data-driven analysis, also known as unsupervised learning, belongs to the field of data mining. The term refers to a set of methods relying mainly on the given data to extract information. In contrast to supervised learning methods, no external source of information, such as labels, is required. However, while prior knowledge about the data is not necessary, it may be helpful to decide on the applied methods and parameters, as well as the interpretation and evaluation of results. In the field of MSI data analysis, the data is often unlabeled and research questions or analysis objectives are not precisely defined or even unknown in advance. Sometimes the exploration of the data is itself the task to be accomplished. In these cases the application of supervised machine learning approaches is impossible. Therefore, this thesis focuses on unsupervised and data-driven analysis methods.

Exploration & Analysis Despite all technological advances in the recent years, there are still three classical approaches, which are predominantly used for the exploration and analysis of MSI data:

1. Manual visual analysis of individual m/z-images, often supported by a kind of m/z-image browser.

2. Cluster analysis

3. Dimension reduction analysis

Cluster Analysis Since clustering is an ill-posed problem, there exists no precise definition for it [30]. However, as a generally valid statement, it can be said that the task of clustering can be described as follows: Given a set of data objects, group them such that objects within the same group are in some way more similar to each other than to objects of other groups.
Formally, a clustering algorithm can be defined as a mapping function. Depending on the use case there are several variants for the definition of such a mapping function. In this thesis, we will stay with the following definition: $\mathbb{F} : \mathbb{R}^d \rightarrow \mathbb{N}$. Each data point x_q is assigned to a unique integer number $\mathbb{F}(x_q)$, which defines the assignment to a cluster. For algorithms which require a predefined number of clusters k, the mapping space is restricted by $\mathbb{F} : \mathbb{R}^d \rightarrow \{0, \ldots, k-1\}$. Otherwise, $\mathbb{F} : \mathbb{R}^d \rightarrow \{0, \ldots, |X|-1\}$, where $|X|$ is the total number of all data points.
As explained in Section 1.2 MSI data features two domains that can be analyzed separately. This also applies to cluster analysis.

Spatial Clustering In the case of spatial clustering, each data point corresponds to an m/z-image $\mathcal{I}_z$, with $z \in \{0, \ldots, Z-1\}$. The features are represented by the intensity values at each pixel $\mathcal{I}_{h,w,z}$. It may be necessary to apply a 2D $\rightarrow$ 1D transformation to the m/z-images ($\mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{|p|}$), which results in a sample-feature matrix $\mathbb{R}^{Z \times |p|}$ or $\mathbb{R}^{Z \times |\rho|}$ if only spectral pixels are considered. After clustering, each cluster consists of multiple m/z-images. An effective method to find a representative of each cluster is to use an aggregation function that combines the information of the clustered images. For most use cases the mean or median function will generate effective results. However, in some use cases, other functions might be more suitable such as maximum or weighted mean. An outline of the spatial clustering workflow is given in Figure 1.6.
Only a few previous works considered the analysis of spatial distributions through spatial clustering. However, these works could already show that this approach can lead to effective results. Examples are Wijetunge et al., 2014, where a Growing Self Organizing Map (GSOM) was used to cluster m/z-images or Alexandrov et al., 2013, where the EM algorithm is used to optimize the parameters for the subsequent Gaussian mixture model clustering of m/z-images.
Although the given examples provide valuable results, this approach is by far not as popular as the spectral clustering.

Spectral Clustering In the case of spectral clustering, each data point corresponds to a mass spectrum $\mathcal{I}_{h,w}$, with $(h, w) \in \rho$. The features are represented by the intensities for each m/z-value $\mathcal{I}_{h,w,z}$. The result is a sample-feature matrix $\mathbb{R}^{|\rho| \times Z}$.

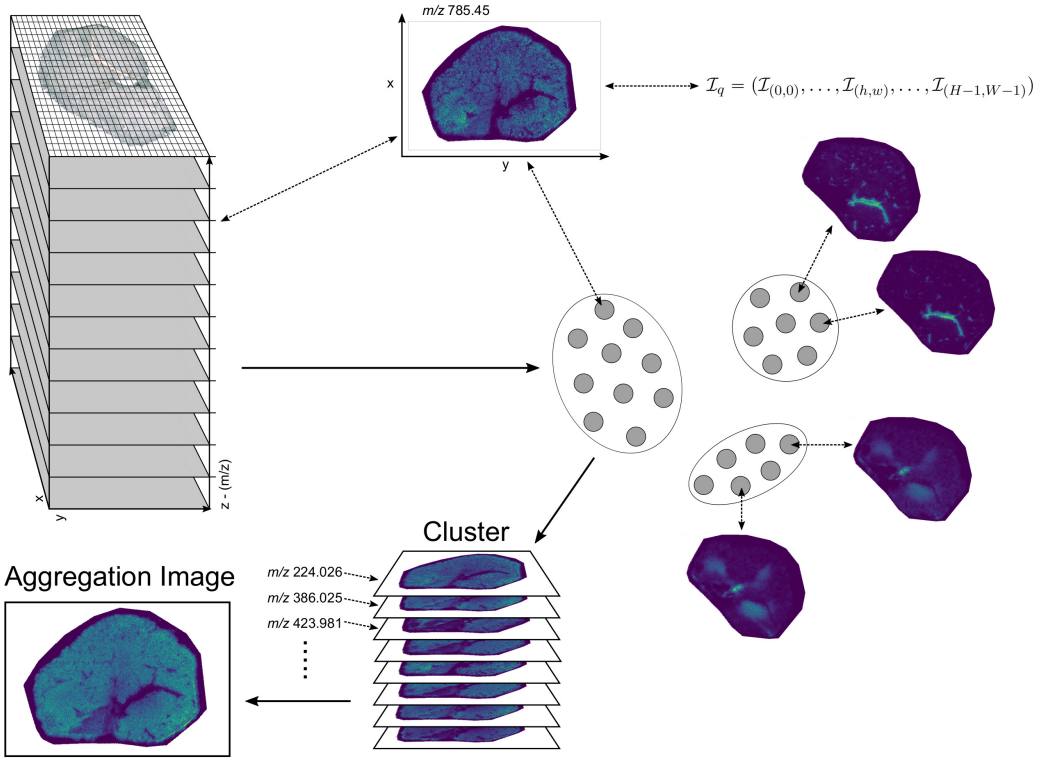


Fig. 1.6.: Spatial clustering workflow – Example of a spatial clustering workflow. Each data point represents an m/z-image. Each cluster consists of multiple m/z-images. An aggregation function can be applied to compute an artificial representative m/z-image for each cluster.

After clustering, each cluster consists of multiple mass spectra, i.e. multivariate spectral pixels. Again, a possible representative of each cluster can be generated by applying an aggregation function. Since the clusters do not overlap, at least as long as no fuzzy clustering scheme was applied, a segmentation map can be used to visualize the clustering result. To create a segmentation map, a unique color is assigned to each cluster. Then, all pixels are colored according to their cluster color. The result is a pseudocolor image of the sample, which visualizes the clustering result on the sample surface.

The clustering of mass spectra and the generation of segmentation maps to visualize the results is a frequently used technique. Examples in previous works include the use of k-Means and ISODATA [73], hierarchical clustering [25], high dimensional discriminant clustering [4], spatially aware clustering [7] and hierarchical hyperbolic self-organizing maps [54].

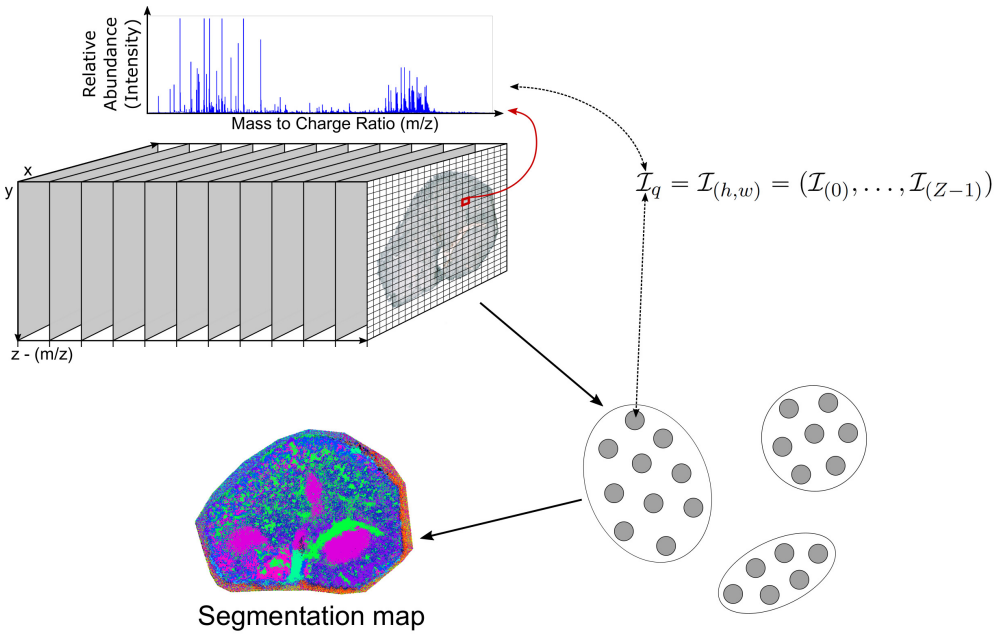


Fig. 1.7.: Spectral clustering workflow – Example of a spectral clustering workflow. Each data point represents a mass spectrum. Each cluster consists of multiple mass spectra. The clustering result can be visualized as a segmentation map.

It is not surprising that clustering and dimension reduction are still frequently used for the exploration and analysis of MSI data. For many use cases, these methods lead to effective results and have an efficient performance. However, the results of these methods are commonly presented in the form of static visualizations or, if interactivity is provided, the interaction options are quite limited. It is important to consider that the results of methods like clustering or dimension reduction can remain very complex. Therefore, an interactive, dynamic and responsive visualization, that allows to filter the amounts of information and to direct the focus to different aspects can be a powerful tool for the exploration of such results and a more detailed analysis. Based on our experience, we believe that the use of visual analysis tools as an integral part of MSI data analysis workflows is still underestimated. For this reason, this thesis will have a strong focus on the topic of visual analysis for MSI data.

1.4 Visualization of Mass Spectrometry Imaging Data

In the scope of this thesis, the term visualization will predominantly refer to interactive information visualization of scientific content. This definition of visualization is

closely connected to the term of *visual analytics*, which evolved from the fields of information visualization and scientific visualization [50]. A very general definition of visual analytics was given in Thomas and Cook, 2005 as "the science of analytical reasoning facilitated by interactive visual interfaces". Keim et al., 2008 extended this definition to an iterative process consisting of data gathering, an interplay between visualization and hypothesis generation, followed by a knowledge gain. Depending on the data and the research question, this process can be executed one-way or iteratively, where the gained knowledge is used in a feedback loop to alter the data source. Similar to Keim et al., 2008 we define three main approaches to visualization, characterized by the task they have to fulfill:

1. **Information presentation:** This term refers to a static visual presentation of already processed information. The process of knowledge gain is already completed and the elaborated results are presented.

2. **Confirmatory visual analysis:** This term can refer either to a static or an interactive visualization, where a hypothesis is already given. This hypothesis provides a starting point for targeted analysis.

3. **Exploratory visual analysis:** This term refers to mostly interactive visualizations without any starting point. Even the precise research question can be unknown. The goal is to explore the data interactively to understand its characteristics and generate knowledge.

When we talk about visual analysis in the context of MSI in this thesis, we refer to a dynamic process that describes the interplay between the data, an interactive visual representation, exploration, hypothesis generation and knowledge gain, illustrated in Figure 1.8. Data refers to the processes of measuring and processing. If the data is given, questions have to be defined. These can be a first hypothesis but it can also focus on the structure or the characteristics of the data. Based on the question, an appropriate (interactive) visual representation can be selected. This leads to a cycle of exploration, knowledge gain and hypothesis generation or refinement. Finally, the knowledge gained from the exploration process can provide an answer to a research question. However, this cycle can also lead to the insight that either a modification to the data source is necessary or that a different visual representation might be better suited. This description already shows that confirmatory and exploratory visual analysis can be strongly interconnected, which is the reason that the visual analysis tools presented in this thesis are located in both.

MSI research often begins without or with a vaguely defined research question, highlighting the importance of interactive visualization tools that allow exploratory

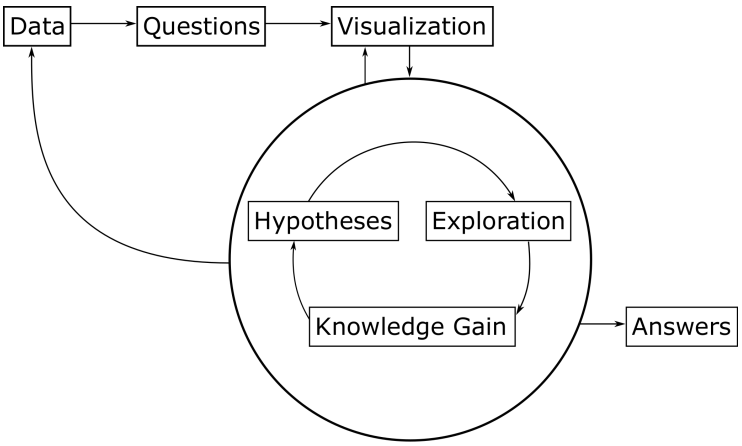


Fig. 1.8.: Visual analysis process – Outline of the process of visual analysis as applied in this thesis. Analysis in the field of MSI often starts with unlabeled data. This data arises questions. Those can be precise research questions or the intention to explore features of the data. Based on the data and the questions, a visualization tool or technique is selected. Working with this visualization leads to a cyclic process that starts with exploration, based on the initial questions. Exploration leads to knowledge. This knowledge can already provide answers or helps to define hypotheses or refine the initial questions. After each step of the cycle, the result can either lead to answers or to the insight that the data source or the visualization tool should be changed.

and confirmatory analysis. The visual analysis tools presented in this thesis can be understood as semi-automated assistive systems. Algorithms are applied to transform MSI data and extract latent information. However, the results of these algorithms are still complex and should be interpreted by experts. Visual analysis tools provide a powerful platform to support these experts in their work.

Design Each visualization tool of this theses was created according to the philosophy: *one visualization for one task*. Instead of building a huge multifunctional software monolith, multiple smaller tools were designed to address specific problems.

The visual analysis tools provided with this thesis follow a few very basic but important design rules. Each tool was developed with Ben Shneiderman's visual information seeking mantra in mind: "*Overview first, zoom and filter, then details-on-demand*" [108]. This mantra emphasizes the importance of the interplay between the need for overview and the need to get details. It also emphasizes the role of data reduction [78]. For low data volumes with low complexity, it would be possible to create an overview by presenting the whole information at once. However, for high volume data, such as MSI data, some form of abstraction is required to reduce the amount of information. The possibility to switch back and forth between overview

and detail allows to efficiently navigate through the vast amounts of information so that analysis does not become unmanageable. The mantra already indicates that data reduction is needed for both, to present an overview and to show details. Powerful methods for data reduction are aggregation, abstraction, zoom and filter [79]. Aggregation and abstraction are well-suited methods to present overviews, while a combination of zoom and filter is suitable to present details at increasingly finer levels.
An additional design objective was to keep every tool simplistic to reduce the amount of chartjunk. The term of chartjunk was introduced by Edward R. Tufte in 1983 [121] and describes the portion of non-data ink or redundant data-ink within a visualization. It should only be used with the purpose to assist orientation. In general, such elements should be reduced as much as possible and if used, they should be muted in relation to the data. In terms of interactive visual analysis tools, the problem of chartjunk can also be transferred to user interfaces. Therefore, the rules for handling chartjunk were incorporated into the design philosophy for our user interface elements. Derived from the development of colorbrewer2[40] and our own experience, we found that learning an interface consists of three critical steps:

1. Learn which interface elements and possible interactions exist.
2. Learn what elements and interactions do.
3. Learn the order in which to use them.

A consistent design decreases the amount of learning time. The reason is that with a consistent design, elements and interactions often behave similarly, which makes it easier to anticipate functionalities [40]. In addition, every visible element stimulates the visual senses, which in turn leads to an additional cognitive load [139]. An intricate user interface increases the cognitive load and reduces intuitiveness, which means that its operation is difficult to learn. As MSI data itself and the information conveyed by the visualization are already highly complex, their exploration and analysis requires a high cognitive effort. This supports the design philosophy that every element not encoding information about the data should be as simplistic as possible. Another aspect that needs to be considered to follow a simplistic design includes the prudent use of negative space. Negative space, also called white space, describes the ink-free area within a visualization. While at the first glance the use of negative space may seem to be a waste of display space, it can be an effective way to convey information like functional and semantic grouping or hierarchies of display elements [135]. The semantic cohesiveness reduces the likeliness to draw false conclusions by combining information of semantically independent display

elements. Negative space also prevents overcrowded panels and can increase the saliency of important display elements, which helps to draw the attention of the user. An increased salience facilitates the users' ability to process the presented information more quickly [138].

Interactivity On an abstract level, interactive visual analysis tools can be divided into the representational component and the interactive component. Both of these components can be closely interwoven [148]. A major advantage of interactive visualizations is the fact that they allow to present and explore a much larger information space than a static visualization could. While interactivity always demands an additional burden on the human attention [78], for highly complex data, such as MSI data, the additional representational power outweighs this disadvantage.
Interactive techniques can be seen as features that provide the ability to directly or indirectly manipulate the presentation of the data [148], which enhances and extends the interpretation capabilities. Some of the most important interaction techniques include dynamic projection, filtering, zooming, link & brush and distortion (like hyperbolic or spherical) [49]. For all these techniques it remains important to maintain the relations between items across all transformations, which emphases the importance of overview & detail and focus & context techniques. Many taxonomies exist to categorize interaction techniques [148]. However, we prefer a data and analysis-driven approach. As a rule of thumb, one should ask: Given a specific type of data and possibly an analysis result, what are the questions that should be answered and which interactions are necessary or useful to support this?
Interactivity should always provide a smooth and fluid user experience. Latency and response time are important factors as delayed actions can strain the patience of the user, which may influence the already stressed attention. Every action that takes longer than one second is no longer perceived as an immediate response [78]. This shows that smooth interactivity can become a problem, especially with high volume data.

Colors and Color Schemes The use of color is one of the most used visual encoding channels. In the literature, various terms can be found to describe the mapping between data values and colors [80]. Common terms are colormap, colorscale, color scheme or pseudo-coloring. In this thesis we will stay with the term of colormap to describe a series of colors that are associated with specified scalars and colormapping for the process that encodes a scalar value with the respective color.
It is important to clarify that there is no perfect colormap for every situation. To be more precise, the selection of a suitable colormap largely depends on the data

type and its features, as well as the task to be performed and the questions to be answered, but also the expected viewers and the presentation medium [77, 97].
Colormaps can be categorized into three main types: sequential, diverging and qualitative [80, 40], while the data to be encoded can be classified either as categorical or as ordered.
In case of categorical data, the only suitable colormap choice would be a qualitative one. This is because these colormaps vary mainly in hue, while saturation and lightness are kept nearly constant. Differences in hue do not automatically imply order. Mixing different "types" of color such as neon and pastel should be avoided, as this could be perceived as some kind of order [40].
In the case of ordered data, both sequential and diverging colormaps are suitable. A key difference between sequential and diverging colormaps is that diverging ones should always be used if the data provides something that divides it into two groups, such as a critical class, a breakpoint or a midpoint. Examples would be a neutral class, a zero value or a mean or median value. The perception of sequential colormaps is often dominated by the difference in lightness. Therefore, they can be either single or multi-hue color sequences, with multi-hue providing better contrast. Diverging colormaps should always be encoded as multi-hue color sequences, where the midpoint is emphasized by a change in hue and lightness so that the data can be divided into upper-end values, break-point and lower-end values [40, 97]. A well-proven concept to achieve an intuitive understanding of the order of a diverging colormap is to use the concept of warm and cold colors. Warm colors are associated with positive activation, making it suitable to represent positive or high values, while cold colors are associated with negative activation, making it suitable for negative or small values [39]. However, in most cases, the selected sequential and diverging colormaps should be encoded by multi-hue and multi-lightness, especially for continuous data, as the human perception is more sensitive to changes in lightness than to changes in hue [77].
The granularity of the different colormap types can be segmented into discrete bins or continuous steps. In the case of categorical data, a segmented granularity should be appropriate. For ordinal data, the use of a continuous granularity will emphasize an ordered characteristic and the use of a segmented granularity will emphasize a discrete characteristic. Also, a continuous granularity is usually the better choice to express quantitative attributes, while a segmented granularity is usually more appropriate to express qualitative attributes [80].
Another factor when choosing a colormap is that it should produce aesthetically pleasing images that match the characteristics of the data and support the message of the visualization. While this may not provide any added value for the encoding, it is likely that, if people have the choice, they will choose the "prettier" map, even if

it offers a slightly worse encoding quality. The colormap should also be intuitively interpretable and allow an intuitive reverse mapping, i.e. it should allow the anticipation of a scalar value based on color or at least allow a rough estimation of a value range. If, after all these recommendations, there is still more than one suitable colormap left, the one that maximizes perceptual resolution should be selected [77, 40].

Some pitfalls have to be considered when using color to encode scalar values. Color has a strong influence on human perception. Consequently, the choice of the colormap can influence the interpretation of the presented data significantly [98] and a poor choice can introduce a bias to certain value ranges. In addition, color is a relative medium, which means that the visual perception of a color can be affected by neighboring colors [136]. Some research even indicates that color can influence the perceived size of objects or areas [97, 21, 118]. Artifacts that might arise present another pitfall [77], such as artificial borders or contours that are not truly present in the data [97]. Some colors also attract more attention than others, which can lead to false conclusions or poor guidance of the user through the visualization. The widely known *Rainbow* colormap is notorious for causing several of these kinds of problems. When dealing with a segmented granularity one has to be careful with the number of classes to hold the colors of different segments distinguishable [40].

Aside from the color channels themselves, the transparency channel can be used for further encoding options. However, transparency needs to be handled with great care as it interferes with all other color channels. A good example of an application is the superimposition of different layers [80].

When encoding data in a visualization it should be considered that there are more encoding channels aside from color, such as size, angle, shape, curvature, texture or motion [80] and for a specific task, such as comparing values, some of these channels might be more suitable than color [137].

Visualization and Visual Analysis of MSI Data As stated before, MSI data is highly complex. This brings several implications for the application of visualization techniques and the development of visual analysis tools.

Not only MSI data itself is highly complex, but also intermediate analysis results like clusters. To support detailed analysis and avoid overcomplication, visual representation of data and results, as well as the design and operation of visual analysis tools, should be as simplistic as possible. In our *one visualization for one task* philosophy, it is highly important to make clear which domain of the MSI data cube is presented and analyzed, i.e. spectral, spatial, or a combination. To achieve this, it can be a good approach to clearly separate interface elements and group them according to

their functionality and presentation purpose. Alexandrov et al., 2013 presents a good example for a multi-display unsupervised visual analysis tool, in which different visualizations are interconnected but also clearly separated from each other.

Interactive data visualization describes the process of generating and presenting an interactive visual representation of the data. However, in the case of MSI, the concept of representation implies more than pure data presentation, since it always includes some kind of transformation in addition to visual encoding. As stated in Section 1.1.1, an m/z-image $\mathcal{I}_z$ consists of multiple measured intensity values from the set of real numbers $\mathbb{R}$. To visualize these m/z-images, a colormapping is required.

A single intensity value describes the measured relative concentration of a given molecule at the measured position. A continuous increase in intensity should be reflected by the colormapping and also be perceived as a continuous increase in magnitude within the visualization. This allows an intuitive reverse mapping [93]. Lightness and saturation have both independently shown that their monotonic increase leads to a perceived monotonic increase in magnitude [98]. At the same time, some molecular distributions show very delicate structures, whose visualization strongly benefits from high contrast. These factors are complemented by the facts that the human visual system is most sensitive to change in lightness and that the intensity values are non-negative real numbers. Consequently, sequential continuous colormaps that change in hue and lightness, such as “viridis”, “inferno” or “fire” [72], are well-fitting options. A very specific exception is the visualization of so-called *component images*. Component images can be interpreted as artificial m/z-images that result from dimension reduction techniques (details to dimension reduction are explained in Section 3.1.1). These images may contain negative values, so a diverging colormap might be useful to distinguish positive and negative valued areas [93]. Examples of multiple different colormaps to visualize m/z-images and their effects can be found in Race and Bunch, 2015.

The value range of an MSI measurement can span several orders of magnitudes. This results in a difficult visualization task and can lead to crucial pitfalls. The correct choice of the colormap is vitally important since a common analysis and evaluation approach for m/z-images is based on the visual inspection of the relative intensities and their distribution pattern. The selected colormap can have a crucial effect on the perceived structures. Thus, some colormaps might draw the attention to interesting structural features, but others may lead to false conclusions [93].

Besides the fact that MSI data already contains non-biological artifacts, such as a high pixel-to-pixel variation and intensity hotspots, a poor selection of the colormap can lead to more artificial artifacts, such as Mach bands [65], overestimation or underestimation of areas [21, 118] and borders that pretend non-existent structures

and morphologies. These artifacts can result in misleading representations of the MSI data and the analysis results, which may eventually lead to false conclusions. The problem of overestimation and underestimation of areas becomes particularly important in the context of spectral clustering when segmentation maps are analyzed. In general, the use of pastel colors supports a more accurate perception of area coverage [93]. A good choice of the colormap can avoid a large number of these artifacts.

Everything mentioned emphasizes the crucial importance of providing high-quality visual analysis tools that use well-suited colormaps and provide intuitively and unambiguously understandable representations of the MSI data and corresponding analysis results. The importance increases even further in the context with MSI advancing towards clinical applications [93].

Another important fact that has to be mentioned is that simplifying the representation is not accompanied by simplifying the data. Highly complex data will always be highly complex and can only be made more accessible. The analysis will still require a high cognitive effort and caution.

1.5 Research Objectives

This thesis deals with the problem of interactive visual exploration of MSI data in combination with unsupervised data-driven analysis. Many different MSI technologies are available, with different technologies serving different purposes. Visual analysis tools should be applicable independent of the MSI technology to increase the value of any type of MSI data. Since this work focuses on the exploratory visual analysis of MSI data, the question of subsequent visualization is already in focus during the selection and development of pre-processing and analysis methods.

To provide the best compatibility with all visual analysis tools of this thesis, a pre-processing pipeline will be presented that transforms the raw data into a visualizable state.

The MSI technology combines a spectral domain with a spatial (imaging) domain and both domains have different requirements for visualization. New visual analysis approaches will be explored and evaluated, for both domains individually and in combination.

For the analysis of the spectral domain, i.e. the comparison of molecular compositions, the focus is on the creation of interactive segmentation maps with high color contrast.

For the analysis of the spatial domain (co-localization), the importance of sample morphology, image resolution and the application of image processing will be investigated, as well as the question of how co-localization can be quantified. Furthermore, a network science-based clustering approach will be presented to detect clusters of m/z-images and a method to approximate regions of interest for clusters of co-localized m/z-images will be proposed.

In addition to the individual analysis of the spatial and the spectral domain, this thesis will also present an example that illustrates how both domains can be combined for an interactive visual analysis.

Data

2

> *For every choice, there is an echo. With each act we change the world.*
>
> **— Dr. Sofia Lamb**
> (Clinical psychiatrist from BioShock 2.)

This chapter provides an overview of the various data sets used in this thesis to evaluate the different methods, algorithms and tools. The data sets are quite different. They originate from different organisms (barley, mouse and human) and were measured with different MSI instruments (MALDI-ToF, MALDI-Orbitrap and AP-SMALDI-Orbitrap). The following data set descriptions provide details about the measurement conditions and basic information before (raw data) and after processing. Furthermore, it contains dimension reduction visualizations to provide a rough overview of the structuring of the feature space in the spatial and spectral domain. The dimension reductions were performed on the entire data set (i.e. before peak picking). Details about the processing procedure and dimension reduction are given in Chapter 3 and Section 3.1.1, respectively.

2.1 Barley Seed

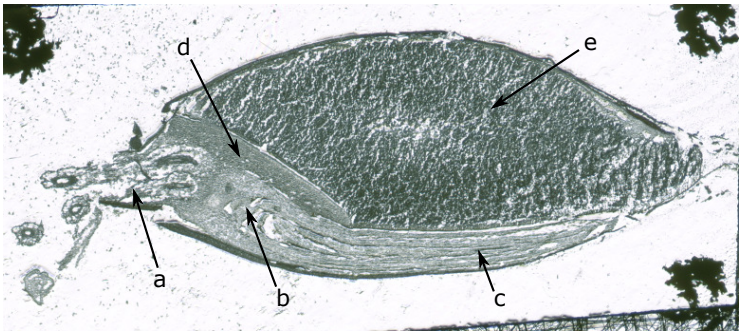
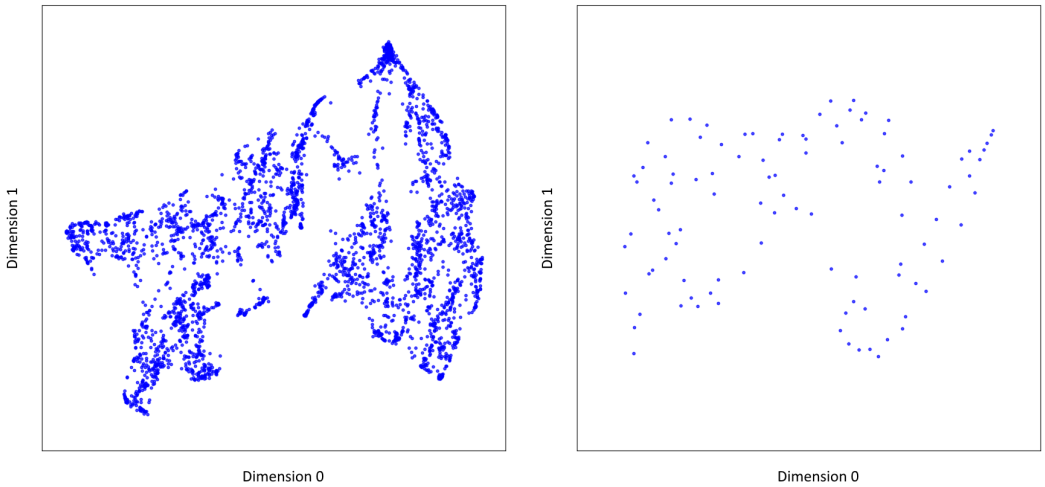


Fig. 2.1.: Brightfield image of the germinating barley seed – The main morphological structures of the seed are marked with arrows [36]: (a) root, (b) center, (c) shoot, (d) scutellum and (e) endosperm. The union of the structures (a),(b) and (c) form the embryo.

The germinating barley seed was embedded in ice. A sample section of $14\ \mu\mathrm{m}$ thickness was cut using a cryostat at $-20\ ^{\circ}\mathrm{C}$. The section was mounted with electric conductive tape on an indium tin oxide (ITO) coated conductive slide (Bruker Daltonics). The sample was coated with the matrix substance 2,5-Dihydroxybenzoic acid (DHB) using an ImagePrep sprayer (Bruker Daltonics). Measurement was performed with a MALDI-ToF/ToF ultrafleXtreme instrument (Bruker Daltonics) in positive ion mode. Using Bruker Daltonics FlexImaging 3.0 software the raster size was set to $100\ \mu\mathrm{m}$. The laser diameter was set to $50\ \mu\mathrm{m}$ and 300 laser shots were accumulated at each raster spot with a laser power within the range of $44\% - 47\%$. Signals were recorded within a detection range of $m/z\ 0 - 3000$, with an ion suppression up to $m/z\ 50$. Manual peak picking (adopted from Gorzolka et al., 2016) resulted in 101 m/z-values between $m/z\ 74.651$ and $m/z\ 2200.687$. The size of each single m/z-image $\mathcal{I}_z$ is 52×111 ($H \times W$), with 3422 spectral pixels $|\rho|$. The resulting data set was stored in the HDF5 storage format with a file size of $2.255\ \mathrm{MB}$. This data set is referred to as $\mathcal{I}^{\mathrm{B}}$. The data set was measured in-house by the research group for proteome and metabolome research.

The use of dimensionality reduction methods on both, the spectral and the spatial domain, is well suited for the early exploration of an MSI data set. Especially the non-linear methods UMAP [75] (Uniform Manifold Approximation and Projection) and t-SNE [64] (t-Distributed Stochastic Neighbor Embedding) are well-suited for visual exploration, because of their topology-preserving nature. Both methods are known to produce low-dimensional embeddings that tend to group data points with similar features in the high-dimensional space.

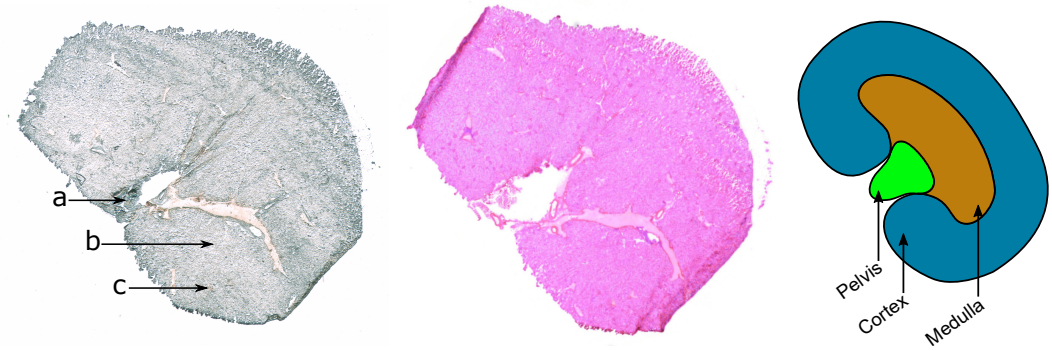
The two-dimensional UMAP embeddings of the barley seed are visualized in Figure 2.2. Figure 2.2 A shows the spectral embedding, meaning that the number of m/z-values is reduced to $Z = 2$, i.e. each data point refers to a two-peak mass spectrum. Similarly, Figure 2.2 B shows a spatial embedding, meaning that the number of spectral pixels is reduced to $|\rho| = 2$, i.e. each data point refers to a two-pixel m/z-image. Both visualizations show areas with increased data point density, which indicates that $\mathcal{I}^{\mathrm{B}}$ features distinctive structural features in the spectral and spatial domain. This leads to the assumption that both domains of $\mathcal{I}^{\mathrm{B}}$ are well-suited for clustering, i.e. clusters are not too hard to compute and will be clearly distinguishable.



(A) Spectral UMAP embedding **(B)** Spatial UMAP embedding

Fig. 2.2.: Two-dimensional UMAP embedding of the entire germinating barley seed data set $\mathcal{I}^{\mathrm{B}}$ – (A) Reduction of the spectral domain, i.e. each data point refers to a two-peak mass spectrum. (B) Reduction of the spatial domain, i.e. each data point refers to a two-pixel m/z-image.

2.2 Mouse Kidney



(A) Brightfield image **(B)** HE-stained image **(C)** Schematic outline

Fig. 2.3.: Brightfield image, HE-stained image (consecutive cut) and schematic outline of the mouse kidney – The main morphological structures are marked with arrows: (a) pelvis, (b) medulla and (c) cortex.

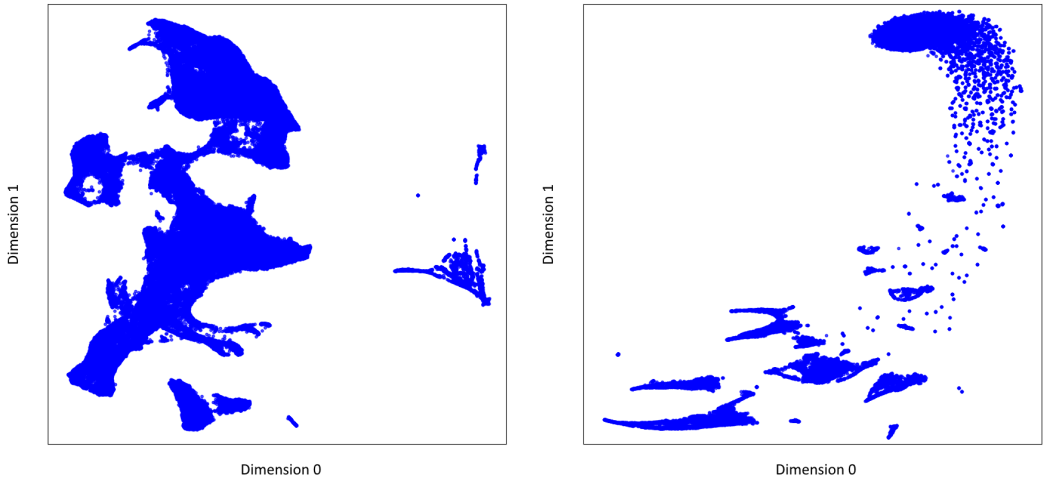
A fresh kidney biopsy tissue sample from a male mouse was rapidly frozen in liquid nitrogen, cut into 10 μm sections in a cryostat (Leica Biosystems) and mounted on ITO coated glass slides with further drying in a desiccator for at least 30 minutes. DHB was prepared as 30 $\mathrm{mg\,mL^{-1}}$ in 50 % Methanol and 1 % trifluoroacetic acid and sprayed onto tissues with a TM-Sprayer (HTX Technologies, LLC) using parameters of 75 $^{\circ}$C , 10 psi, 0.1 $\mathrm{mL\,min^{-1}}$, 1200 $\mathrm{mm\,min^{-1}}$ and 8 passes in a crisscross pattern with a 3 mm spacing.

Measurement was performed with a MALDI-Orbitrap instrument (Q Exactive Plus; Thermo Scientific with MALDI/ESI injector; Spectroglyph, LLC) in positive ion mode. The automatic gain control (AGC) target value (ion population in the mass analyzer) was fixed with an injection time of 250 ms. Diode laser current was set to 2.0 Amps and the laser repetition rate was 500 Hz. The spatial resolution was 20 μm and the spectral resolution was fixed to 70000 bins. Raw data and position data were aligned and exported to imzML with Image Insight Software (version 12068). Hematoxylin and Eosin (HE) staining was applied to the subsequent tissue section. Ready-to-use solutions of Mayer's acidic hemalaun and aqueous 0.5 % eosin G solution (Carl Roth) were filtered before usage. The tissue section was fixed in 100 % methanol (2 min), followed by a washing step (10 dips) with demineralized water (dH2O). The acidophilic structures of the tissue were stained with hemalaun solution (6 min). A second washing step with dH2O followed before blueing under running tap water (10 min). Aqueous 0.5 % eosin G solution was used after a short washing step with dH2O to counterstain the basophilic structures (8 s). Further washing and

differentiation steps were performed with fresh 100 % ethanol (2 x 10 dips). The slide was cleared in xylene and covered using a synthetic mounting medium.

The raw data was stored as imzML, with a combined file size of 21.73 GB(274 MB (imzML) and 21.4 GB (ibd)). After processing, the size of the corresponding HDF5 was reduced to 10.5 GB. The stored data set has a dimensionality ($H \times W \times Z$) of $425 \times 484 \times 47855$, with 154290 spectral pixels $|\rho|$ and an m/z-value range of m/z 100.06901 to 1499.6912. We found that the instrument software stored some error induced artificial spectra. These spectra were all stored to the first spatial position, i.e. at pixel position $(0, 0)$, and contain no relevant information. An artifact profile spectrum was computed (arithmetic mean spectrum) and subtracted from all other spectra. Then, all artifact pixels were removed, which lowered the remaining number of spectral pixels to $|\rho| = 147615$. A set of matrix spectra was interactively selected to compute a matrix profile spectrum. To reduce the influence of matrix characteristic m/z-values on the mass spectra and the m/z-images, the matrix profile spectrum was subtracted from all other spectra. Peak picking was executed with an intensity threshold of 0.04. The peak picking threshold was determined manually with the objective to include a diversity of different interesting molecular distributions (m/z-images), such as morphologically relevant or visually distinct ones, while avoiding the presence of too many less interesting molecular distributions, such as noisy, empty or redundant ones. A thresholding method was used to select the peaks because it is part of the pre-processing pipeline, which will be introduced later in Chapter 3. Furthermore, the idea was to reconstruct a more realistic application scenario in which it is likely that some less interesting molecular distributions will be included in the data set and to reduce the bias caused by the selective selection of individual peaks. Deisotopes were approximated with a range between 0.8 and 1.2 and removed, resulting in 100 m/z-values between a range of m/z 104.107006 and 1120.585648 and a file size of 90.9 MB. To improve the contrast of the m/z-images and reduce the intensity value variance, the data set was transformed using a square-root transformation. This data set is referred to as $\mathcal{I}^{\mathrm{K}}$. The data set was measured in-house by the research group for proteome and metabolome research.

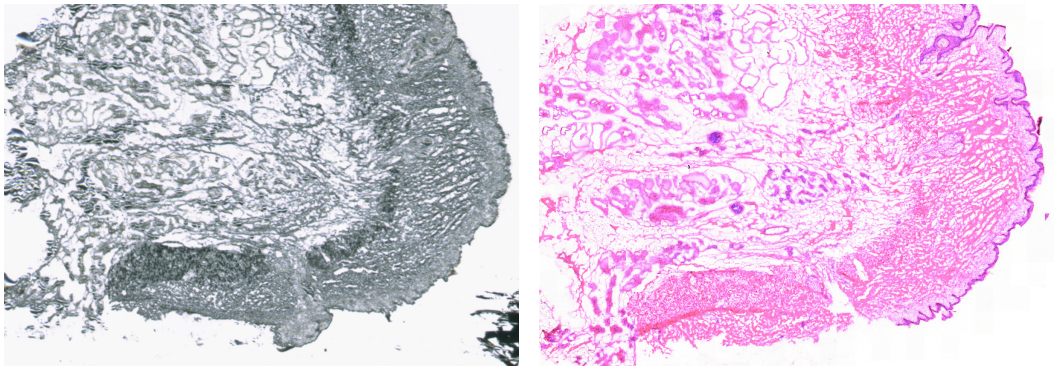
Similar to Figure 2.2 in Section 2.1, Figure 2.4 shows two UMAP embeddings, one for each domain of the mouse kidney data set. Both embeddings show areas with increased data point density, which indicates the existence of distinctive structural features in both domains. Like for $\mathcal{I}^{\mathrm{B}}$ before, this leads to the assumption that both domains of $\mathcal{I}^{\mathrm{K}}$ are well-suited for clustering, i.e. clusters are not too hard to compute and will be clearly distinguishable.



(A) Spectral UMAP embedding **(B)** Spatial UMAP embedding

Fig. 2.4.: Two-dimensional UMAP embedding of the entire mouse kidney data set $\mathcal{I}^{\mathrm{K}}$ – (A) Reduction of the spectral domain, i.e. each data point refers to a two-peak mass spectrum. (B) Reduction of the spatial domain, i.e. each data point refers to a two-pixel m/z-image.

2.3 Human PXE Skin



(A) Brightfield image **(B)** HE-stained image

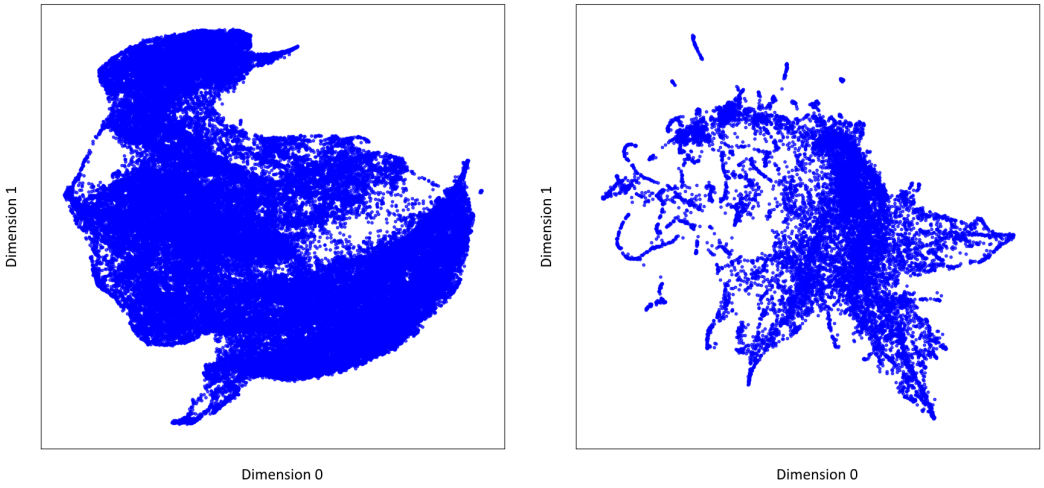
Fig. 2.5.: Brightfield image and HE-stained image (consecutive cut) of the human skin – In both images, at least two differently structured tissue areas are visible, i.e. more "densely" and less "densely" structured tissue.

A human skin tissue sample diseased by Pseudoxanthoma elasticum (PXE) was rapidly frozen in liquid nitrogen, cut into 10 μm sections in a cryostat (Leica Biosystems) and mounted on ITO coated glass slides with further drying in a desiccator for at least 30 min. DHB was prepared as 30 $\mathrm{mg\,mL^{-1}}$ in 50 % Methanol and 1 % trifluoroacetic acid and sprayed onto tissues with a TM-Sprayer (HTX Technologies, LLC) using parameters of 75 °C, 10 psi, 0.1 $\mathrm{mL\,min^{-1}}$, 1200 $\mathrm{mm\,min^{-1}}$ and 8 passes in a crisscross pattern with a 3 mm spacing. HE-staining was applied to the subsequent tissue section. Ready-to-use solutions of Mayer's acidic hemalaun and aqueous 0.5 % eosin G solution (Carl Roth) were filtered before usage. The tissue section was fixed in 100 % methanol (2 min), followed by a washing step (10 dips) with demineralized water (dH2O). The acidophilic structures of the tissue were stained with hemalaun solution (6 min). A second washing step with dH2O followed before blueing under running tap water (10 min). Aqueous 0.5 % eosin G solution was used after a short washing step with dH2O to counterstain the basophilic structures (8 s). Further washing and differentiation steps were performed with fresh 100 % ethanol (2 x 10 dips). The slide was cleared in xylene and covered using a synthetic mounting medium.

Measurement was performed with a MALDI-ToF (ultrafleXtrem; Bruker) in positive ion mode, collecting 350 laser shots per pixel and a laser frequency of 1000 Hz. The laser global offset was set to 71 % and laser energy was set to 60 %. The detector voltage was set to 1200 V. The spatial resolution was 20 μm and the laser diameter was "medium". The data was exported to imzML by flexImaging 4.1.

The raw data was stored as imzML, with a combined file size of 11.32 GB(110 MB (imzML) and 11.2 GB (ibd)). After processing, the size of the corresponding HDF5 was reduced to 6.89 GB. The stored data set has a dimensionality ($H \times W \times Z$) of $278 \times 396 \times 20000$, with 75210 spectral pixels $|\rho|$ and an m/z-values range of m/z 99.98101 to 1149.98816. The measurement contains a dedicated matrix measurement field. A set of matrix spectra from a dedicated matrix field was interactively selected to compute a matrix profile spectrum. To reduce the influence of matrix characteristic m/z-values on the mass spectra and the m/z-images, the matrix profile spectrum was subtracted from all other spectra. Then, the pixels of the matrix field were removed, which lowered the number of spectral pixels to $|\rho| = 75042$. Peak picking was executed with an intensity threshold of 0.052. The peak picking threshold was determined manually, with the same objective as discussed for $\mathcal{I}^{\mathrm{K}}$. Deisotopes were approximated with a range between 0.8 and 1.2 and removed, resulting in 51 m/z-values between the range m/z 100.13001 and 907.75663 and a file size of 9.95 MB. To improve the the contrast of the m/z-images and reduce the intensity value variance, the data set was transformed using a successive logarithmic and square-root transformation. This data set is referred to as $\mathcal{I}^{\mathrm{S}}$. The dataset was measured in-house by the research group for proteome and metabolome research.

Similar to Figure 2.2 in Section 2.1, Figure 2.6 shows two UMAP embeddings, one for each domain of the human skin data set. In this case, both embeddings appear to be less structured and do not reveal any distinctive groupings, which indicates that $\mathcal{I}^{\mathrm{S}}$ is more "*complex*" in its molecular and structural features than the other data sets. This leads to the assumption that the data set exhibits fewer distinctive structural features in both domains or that these features are of a less regular nature. Consequently, the computation of clusters in both domains might be a hard task.



(A) Spectral UMAP embedding **(B)** Spatial UMAP embedding

Fig. 2.6.: Two-dimensional UMAP embedding of the entire human skin data set $\mathcal{I}^{\mathrm{S}}$ – (A) Reduction of the spectral domain, i.e. each data point refers to a two-peak mass spectrum. (B) Reduction of the spatial domain, i.e. each data point refers to a two-pixel m/z-image.

2.4 Mouse Urinary Bladder

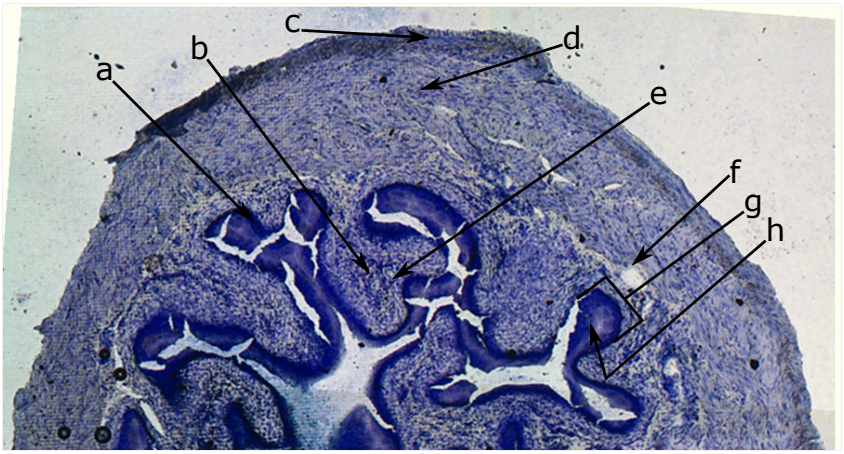


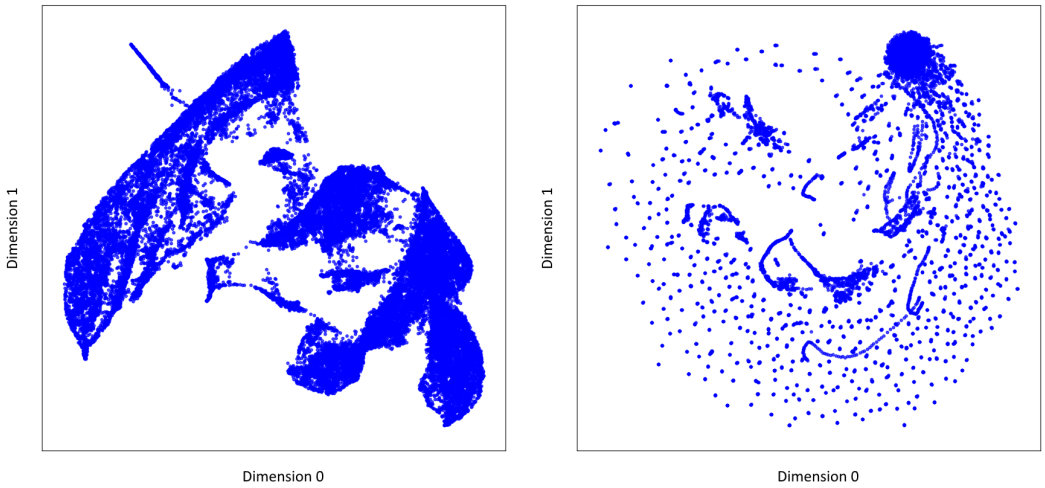
Fig. 2.7.: Toluidine blue stained image of the mouse urinary bladder – The morphological structures adopted from Römpp et al., 2010 are marked with arrows: (a) basal layer, (b) lamina propria, (c) adventitial layer, (d) detrusor muscle, (e) myofibroplasts, (f) blood vessel, (g) urothelium and (h) umbrella cells.

A fresh mouse urinary bladder was immediately frozen in liquid nitrogen. The sample was cut into sections of 20 μm thickness with a cryotome (HM500, Microm). The sections were mounted on a conductive ITO coated glass slide and stored at -80 $^{\circ}$C for a maximum of one week. Before measurement, the section was brought to room temperature in a desiccator for 30 minutes. DHB was applied as a matrix substance using a pneumatic sprayer. Measurements were performed with an AP-SMALDI ion source, attached to a linear ion trap/Fourier transform orbital trapping MS (LTQ Orbitrap Discovery, Thermo Scientific) in positive ion mode. The laser was focused to a diameter of 5 to 10 μm with a resulting pixel size of 10 μm . Further details can be found in Römpp et al., 2010

The raw data is available as imzML, with a combined file size of 830.5 MB(53.5 MB (imzML) and 777 MB (ibd)). After processing, the size of the corresponding HDF5 was reduced to 302 MB. The stored data set has a dimensionality ($H \times W \times Z$ of $134 \times 260 \times 8562$), with 34840 spectral pixels $|\rho|$ and an m/z-values range of m/z 400.33903 to 996.55968. A set of matrix spectra was interactively selected to compute a matrix profile spectrum. To reduce the influence of matrix characteristic m/z-values on the mass spectra and the m/z-images, the matrix profile spectrum was subtracted from all other spectra. Peak picking was executed with an intensity threshold of 0.0197. The peak picking threshold was determined manually, with the same objective as discussed for $\mathcal{I}^{\mathrm{K}}$. Deisotopes were approximated with a range between 0.8 and 1.2 and removed, resulting in 150 m/z-values between

the range m/z 406.95702 and 988.15668 and a file size of 26.4 MB. To improve contrast of the m/z-images and reduce the intensity value variance, the data set was transformed using a logarithmic transformation. This data set is referred to as $\mathcal{I}^{\mathrm{U}}$. The data set is publicly available at the PRIDE database [126] (`https://www.ebi.ac.uk/pride/archive/projects/PXD001283`).

Similar to Figure 2.2 in Section 2.1, Figure 2.8 shows two UMAP embeddings, one for each domain of the mouse urinary bladder data set. Both embeddings show areas with increased data point density, which indicates the existence of distinctive structural features in both domains. However, Figure 2.8 B also shows multiple well-spaced single data points. In this quantity, this could be an indication of noise. An in-depth m/z-image inspection confirmed that these data points belong to noisy m/z-images.



(A) Spectral UMAP embedding **(B)** Spatial UMAP embedding

Fig. 2.8.: Two-dimensional UMAP embedding of the entire mouse urinary bladder data set $\mathcal{I}^{\mathrm{U}}$ – (A) Reduction of the spectral domain, i.e. each data point refers to a two-peak mass spectrum. (B) Reduction of the spatial domain, i.e. each data point refers to a two-pixel m/z-image.

Pre-Processing

3

Parts of this Chapter are based on:
Fast and Visual Exploration of Mass Spectrometry Images with Interactive Dynamic Spectral Similarity Pseudocoloring
`Code: https://github.com/Kawue/provim/`

> *The whole is other than the sum of the parts.*
>
> **— Kurt Koffka**
> (Gestalt psychologist)

3.1 Motivation

The raw data output of an MSI measurement suffers from a multitude of different artificial and biological artifacts, as well as non-biological variations. Since all these artifacts and variations carry no biologically relevant information, they will be collectively referred to as artifacts. To explore and reveal biologically relevant information through subsequent downstream analysis, the impact of artifacts should be reduced to a minimum. Artifacts can originate from a lot of different sources such as variations in instrument parameter settings, environmental conditions, variation in ionization and detector efficiency, sample preparation protocols and inhomogeneity of external matrix deposition and sample morphology, mainly due to variations in thickness. If the experiment aims for comparative analysis rather than a single sample analysis, different samples and measuring instruments constitute another major source of non-biological variation. The application of appropriated pre-processing methods is necessary to reduce the impact of these artifacts.

The most basic pre-processing steps are often performed by the instrument software directly. This includes methods like baseline correction and mass spectrum binning into equally spaced bins.

Baseline correction In theory the baseline of each mass spectrum is equal to zero. However, with many MSI techniques, each mass spectrum measurement begins with an increased baseline, which slowly decreases during the measurement process. Any baseline above zero is a serious signal distortion, which not only falsifies the measured intensities but also negatively impacts any subsequent signal (pre-)processing. For this reason, it is crucial to apply some form of baseline correction. If no baseline correction is integrated within the instrument software, this has to be the first step of every pre-processing pipeline. For all data sets in this thesis, the baseline was already corrected.

3.1.1 Dimension Reduction

The projection of high-dimensional MSI data into a low-dimensional space through dimension reduction methods is an important aspect of our pre-processing pipeline and is frequently used for MSI analysis in general. Thus, the concept of dimension reduction will be briefly introduced below. Dimension reduction can also be quite helpful for quality control purposes and a rough early overview of the structure of the feature space of the data.

3.1.2 High Dimensional Data

Despite the advantage of providing huge amounts of information, high dimensional data comes with a major disadvantage, a phenomenon termed as the *"curse of dimensionality"*. The term was originally coined by Richard E. Bellman[9, 10]. Nowadays, it refers to any problem in data analysis caused by the presence of a huge number of features.

A problem that that is usually accompanied by an increasing number of features, i.e. an increasing dimensionality of the data, is an increase of the sparsity. Sparse data can lead to problems in the application of statistical methods and statistical significance. Another problem is the increase in combinatorics, referred to as *"combinatorial explosion"*, which eventually leads to runtime and memory issues. Furthermore, pairwise distances that return a non-negative real number tend to become relatively uniform with increasing dimensionality, which poses a problem for cluster analysis. The problem for cluster analysis can be briefly explained as follows: Given the distances to the nearest and farthest neighbor of a high-dimensional data

point δ_{min} and δ_{max}, the results of Beyer et al. 1999 showed that with increasing dimensionality d the difference between δ_{min} and δ_{max} approaches zero:

$$\lim_{d \to \infty} \frac{\delta_{\text{max}} - \delta_{\text{min}}}{\delta_{\text{min}}} = 0 \tag{3.1}$$

This means that with increasing dimensionality all data points converge to the same distance from any selected data point. A geometrical example for an easier understanding was given in Steinbach et al. 2004: imagine a hypersphere centered at an arbitrary selected high-dimensional data point x_q, with a radius given by the distance to its nearest neighbor δ_{min}. Because the distance between the nearest and the farthest neighbor in high-dimensional space is very small, expanding the radius only slightly will include many more data points. The problem increases with increasing dimensionality, as the difference becomes smaller. Consequently, the concept of a nearest-neighbor becomes increasingly meaningless in high-dimensional spaces. Since many clustering algorithms rely on the concept of nearest-neighborhoods the stated problem can have a serious impact on the cluster analysis results. The application of dimension reduction methods is one option to address this problem [13, 2, 114].

Dimension reduction Dimension reduction can be described as a function $f(x_q) = y_q$ that calculates a projection. This function is referred to as embedding projection. The embedding projection projects a high-dimensional data point $x_q \in \mathbb{R}^d$ to a low-dimensional representation $y_q \in \mathbb{R}^{d'}$. Thus, $\mathbb{R}^{d'}$ represents a low-dimensional space based on $\mathbb{R}^d$, where $d' \ll d$. The low-dimensional space $\mathbb{R}^{d'}$ is referred to as embedding space and the projected data is referred to as embedding.
The axes of the coordinate system that describes the computed embedding space are referred to as embedding axes. A common objective of the various dimension reduction methods is to compensate some of the problems inherent to the high-dimensional space while preserving some of the properties of the data. A frequently used application case for dimension reduction is the projection of high-dimensional data into a two- or three-dimensional space $\mathbb{R}^d \to \mathbb{R}^{d'}$, where $d' = 2 \vee d' = 3$. The reason is that two- and three-dimensional spaces allow a straight forward visualization, which in turn allows a manual visual exploration.
Additionally, the terms sub-embedding and m/z-embedding-image are briefly defined, because they are needed later on.
A sub-embedding is defined as the projection of the embedding on a subset of embedding axes from the embedding space. For example, given a dimension reduction from $\mathbb{R}^d \to \mathbb{R}^{d'}$, with $d' > 2$ and the task to visualize the embedding on two selected

axes $\mathbb{R}^{d''}$ as a scatterplot. Then, the projection of the embedding in $\mathbb{R}^{d''}$ is defined as a sub-embedding. Naturally, there can be various sub-embeddings, because any combination of axes less than d' can be used to build $\mathbb{R}^{d''}$.
For a dimension reduction of the spectral domain, each data point of the embedding represents an embedded mass spectrum, which is associated with a specific pixel. Thus, an m/z-embedding-image represents the intensity value distribution pattern of the embedding, projected on one specific embedding axis.

In general, dimension reduction methods can be divided into two categories: linear methods and non-linear methods. In the following, we will provide a brief introduction to two methods:

1. Principal Component Analysis (PCA), which is a linear projection method.

2. Uniform Manifold Approximation and Projection (UMAP), which is a non-linear projection method.

Principal Component Analysis The principal component analysis (PCA) is probably the most known and commonly used dimension reduction method. The embedding space is characterized by two conditions. First, all axes are pairwise orthogonal and therefore linearly uncorrelated and second, respecting the orthogonality, the direction of each axis points along the maximum variance of the data.
In the context of PCA, the embedding axes are usually called principal components. They are determined by computing the eigenvectors of the covariance matrix of the data. By nature, eigenvectors are pairwise orthogonal, which ensures that the first property is respected. To respect the second property, the eigenvectors are sorted in descending order according to their associated eigenvalues. This order defines the order in which the embedding axes are selected. The reason for this is given by the property that the higher the eigenvalue of an eigenvector, the higher is the variance covered by this eigenvector.
Two frequently used coefficients are the amount of variance explained and the variance explained ratio. The computed eigenvalues (λ_n) describe the amount of variance covered by each eigenvector. Consequently, the variance explained ratio is the ratio of a single eigenvalue to the sum of all eigenvalues $\frac{\lambda_n}{\sum_n \lambda_n}$.
If the objective is to reduce the dimensionality of a data set, at some point the question of how many principal components should be selected will arise. The answer largely depends on the analysis objective. However, frequently used choices are either two or three, if visualization is required, or enough to cover a certain amount of total variance explained, such as 90%, 95% or 99%.

Uniform Manifold Approximation and Projection The Uniform Manifold Approximation and Projection (UMAP) is a rather young method (presented in 2018 by McInnes et al.) and mathematically way more complex than PCA. Since details are not relevant in this thesis, this paragraph will only give a brief overview of the idea behind UMAP to provide an intuition about how the algorithm works.
UMAP is a non-linear dimension reduction method that constructs a high-dimensional nearest-neighbor graph. The method aims to construct a low-dimensional nearest-neighbor graph that is structurally as similar as possible to its high-dimensional counterpart. This is achieved with a fuzzy simplex cover approach. A fuzzy simplex cover can be understood as a weighted graph, where the weight of each edge represents the likelihood that two data points are connected. In this context, connected means that a ball with a defined radius is placed around each data point. If the balls of two data points overlap, these two data points are considered connected. If the radius is too small, many data points and groups of data points will be isolated. This discards information about their relationships within the global structure. If the radius is too large, everything will be connected. This discards information about local structures. To solve this problem the local radius for each data points' ball is based on the distance to the data points' k'th nearest-neighbor. This binary relation is converted into a fuzzy relation by decreasing the likelihood of two data points being connected with increasing distance. To ensure that each data point is connected to at least one other data point the fuzzification begins beyond each data points' nearest-neighbor. The low-dimensional graph representation is optimized by using cross entropy as an objective function and stochastic gradient descent as an optimizer [75, 123].
UMAP ensures that the projection preserves the local structure in balance with the global structure. The method also has the advantage that it can be used with virtually any metric [122, 123].

3.1.3 Using Embeddings to Explore Regions of Interest

A convenient feature of dimension reduction methods is that if the embedded data points are similar in a certain way, they may form groups, i.e. the embedding may reveal clusters. Whether clusters become visible and how they have to be interpreted depends on the data, the dimension reduction method used and the selected dimensions. A two-dimensional visualization as a scatterplot is an effective method for the exploratory analysis of potential clusters in embeddings. This can be done by either using a two-dimensional embedding directly or by visualizing multiple two-dimensional scatterplots for the various two-dimensional sub-embeddings.

However, using an interactive visual analysis tool that allows to dynamically change embedding axes and creates different sub-embedding is way more efficient than using multiple static scatterplots.
Similar to cluster analysis, which was discussed in Section 1.3, dimension reduction methods can be applied to both domains of an MSI data set. However, applying a dimension reduction method to an MSI data set usually requires a representation of the three-dimensional data cube as a two-dimensional sample-feature matrix. Details how to create sample-feature matrices for both domains are given in Section 1.3.
To facilitate the interpretation of potential clusters in the context of regions of interest analysis, it is helpful to provide a function that presents a selected set of data points in the context of either the spatial or spectral domain of the MSI data set.

Spatial dimension reduction Given a sample-feature matrix $\mathbb{R}^{Z \times |\rho|}$, each data point corresponds to an m/z-image $\mathcal{I}_z$, with $z \in \{0, \ldots, Z-1\}$. The features are represented by the intensity values at each pixel. The application of a dimension reduction method reduces the number of spectral pixels (mass spectra), while the number of m/z-values remains the same. Consequently, selecting a set of data points equals the selection of a set of m/z-values.

Spectral dimension reduction Given a sample-feature matrix $\mathbb{R}^{|\rho| \times Z}$, each data point corresponds to a mass spectrum $\mathcal{I}_{h,w}$, with $(h, w) \in \rho$. The features are represented by the intensities for each m/z-value. The application of a dimension reduction method reduces the number of m/z-values, while the number of mass spectra remains the same. Consequently, selecting a set of data points equals the selection of a set of mass spectra. By using the (h, w) coordinates associated with each mass spectrum, a selection can be visualized within the context of the spatial domain. If the focus is purely on spatial relation, a representation as a binary image provides an effective visualization. By allowing an interactive selection of the embedded data points, the spatial relationship of mass spectra that are organized in clusters can be analyzed effectively and efficiently, which will be relevant in the course of this chapter.

3.2 Pre-Processing Pipeline

At present, various MSI measurement techniques are available and actively used, each with its advantages, disadvantages and application purpose. With the increasing number of different MSI instruments, it is important that pre-processing methods and pipelines provide the flexibility to handle the output of a variety of different MSI techniques and are not excessively tailored to any instrument or vendor-specific data properties. Since the spatial and spectral resolution of the various instruments can vary greatly, it is also important that the applied methods and algorithms do not have excessive demands on computational resources.

In some application areas, like clinical pathology, where fast and reliable MSI instruments such as MALDI-ToF are the preferred. In other areas, like medical research pathology, a wider range of MSI instruments are applied, for example, MALDI-Orbitrap. This thesis also uses datasets that were measured with different instruments, including MALDI-ToF, MALDI-Orbitrap and AP-SMALDI-Orbitrap.

Usually, MSI data is stored in the vendor-independent raw formats ibd and imzML [105]. However, due to the high dimensionality and volume of the data, specialized pre-processing is required to prepare the raw data for downstream analysis and to obtain accessible and interpretable visualizations [43]. Such a pre-processing should include signal alignment, normalization [31], variance stabilization and peak picking. Another important step that is often neglected is the removal or reduction of tissue-unspecific m/z-signals, e.g. matrix-related m/z-signals. In this context, the term m/z-signal refers to a measured intensity value within a mass spectrum that is associated with a specific m/z-value.

In recent years, several software packages have been proposed for the pre-processing of MSI data. This includes a variety of free tools, such as SpectralAnalysis [94], MSiReader [15], BioMap (Novartis), Datcube Explorer [51], rMSI [95], Cardinal [11] and pyBASIS [125], but also commercial tools such as SCiLS (Bruker Daltonics), Xcalibur/ImageQuest (Thermo Fisher Scientific) and High Definition Imaging (HDI, Waters Corporation).

Some of the aforementioned tools use methods such as dimensional reduction and clustering to automatically identify and exclude matrix-related mass spectra (matrix pixels) for further (statistical) analysis. Other tools use these methods to identify matrix-related m/z-values and some provide intensity value transformations to enhance weak signals and flatten strong signals, e.g. logarithmic or square-root transformations. The application of such transformations can cause some problems.

They compress the range and variance of intensity values and are susceptible to the increase of noise signals. The automated identification methods are either limited to the exclusion of matrix pixels or the return of indices of matrix pixels or matrix-related m/z-values. Providing only an identification of matrix pixels or matrix-related m/z-values requires user-written solutions for the reduction or removal. Furthermore, since the detection methods are fully automated, they do not provide the possibility to manually investigate and correct the results of this sensitive pre-processing step. None of the tools mentioned above allow accurate identification of tissue-unspecific m/z-signals and a subsequent reduction of these signals in every mass spectrum of the data set. Although, such a reduction can be useful to improve the signal contrast in subsequent visualizations.

This chapter introduces the basic pre-**pro**cessing pipeline for the **v**isualization and multivariate analysis of **M**SI data (ProViM). ProViM is a lightweight pipeline that satisfies all the requirements mentioned above. It was developed to prepare MSI data for multivariate statistical analysis and visualization, as well as to ensure the compatibility with all visual analysis tools presented in this thesis.

ProViM has a non-monolithic design, which means that it consists of various individual modules that can be executed in succession. It is based on universally applicable and cost-efficient algorithmic approaches. An outline of the modular construction is provided in Figure 3.1. Different from the other tools that were mentioned above, it provides a module that allows the interactive selection and reduction of tissue-unspecific signals. It also provides a module for interactive peak picking and the approximation and removal of isotopes. For most data sets, it is very likely that the removal of tissue-unspecific signals and the selection of peaks will benefit greatly from manual tuning, as these methods are very sensitive. Consequently, it is always advisable to use ProViM interactively to have full control over these sensitive pre-processing methods through manual intervention. However, situations may arise where time is a limiting factor, e.g. in a large series of experiments. For these situations, each interactive ProViM module offers alternative algorithmic approaches that allow a fully automated execution. To facilitate the execution and ensure the independence of the operating system, ProViM is prepared to be executable in a Docker container.

ProViM was used to process the data sets of the mouse kidney ($\mathcal{I}^{\mathrm{K}}$), the human skin ($\mathcal{I}^{\mathrm{S}}$) and the mouse urinary bladder ($\mathcal{I}^{\mathrm{U}}$).

The following sections will provide in-depth explanations about each processing step.

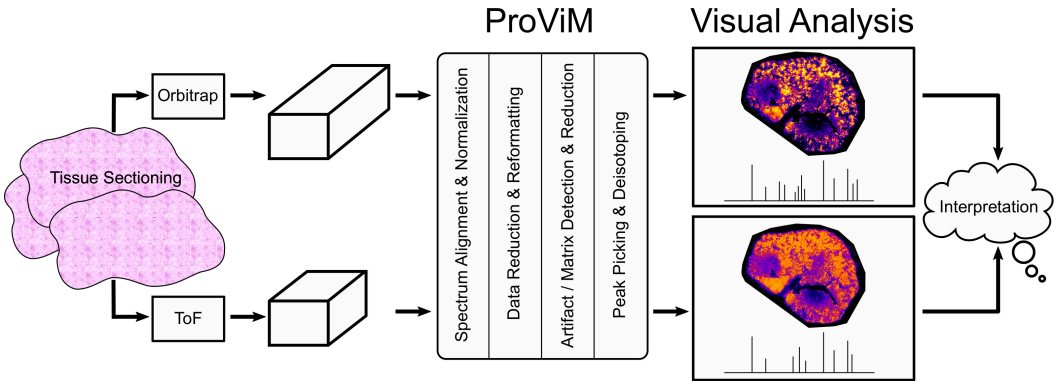


Fig. 3.1.: ProViM workflow – Illustration of the ProViM pipeline using the MSI instruments ToF and Orbitrap as examples. After data generation, the pipeline successively executes four pre-processing steps: 1. spectrum alignment and normalization, 2. data reformatting, 3. artifact and matrix detection and reduction and 4. peak picking and deisotoping. After pre-processing, the data is ready to be explored and analyzed with a visual analysis tool of this thesis.

3.2.1 Alignment & Normalization

Mass shifts are an inherent problem of MSI technology. They can lead to complications when a mean or median spectrum is created during molecule identification and overestimate the number of peaks during peak picking. The mass shift for a single spectrum can be non-linear along the m/z-axis and the shift structure can be different for any spectrum, which makes the problem even harder. To reduce this type of artificial error an alignment method (also referred to as mass drift correction) should always be applied. Another inherent problem of MSI measurements are systematic differences in the overall measured intensity of ions, which applies to mass spectra at different spatial locations within a single sample and to measurements between samples. These systematic differences can have various reasons. For MALDI-MSI some of the reasons include heterogeneity in matrix application, differences in tissue thickness and variation in desorption, ionization or detector efficiencies [31, 125]. For both cases, this causes non-biological (artificial) variability, which should be reduced by normalization methods to not interfere with any downstream analysis.

For the alignment and normalization of all spectra ProViM integrates the corresponding functions of another recently released software package called pyBASIS [125]. The algorithms for both methods are fast, lightweight and independent of the MSI instrument.

The alignment method of pyBASIS works in a reference-based fashion. A kernel-based clustering approach is applied to align identical or similar ion species to a common m/z-vector. For comparative analysis approaches, a single alignment

is usually computed for all samples. For samples measured with different MSI instruments, this poses a special problem, as different MSI instruments can have different spectrum properties. Especially because of differences in accuracy and resolution, it should be carefully reconsidered whether an alignment is applicable or whether it could introduce artificial errors.

To reduce the non-biological variability, an intra-normalization method is applied. The intra-normalization of pyBASIS normalizes each spectrum with a spectrum specific scaling factor. For comparative studies, the entirety of all measured spectra is susceptible to non-biological variations. If the measurements are performed with the same MSI instrument the inter-normalization method can be applied in addition to the intra-normalization. However, if the measurements were performed with different MSI instruments, it should be carefully reconsidered, whether inter-normalization is applicable or whether it could introduce artificial errors. The inter-normalization of pyBASIS normalizes each spectrum of a sample with a sample-specific scaling factor.
Both, the intra- and inter-normalization provide three different approaches to compute the scaling factors, which are the mean ion current, the median ion current or the median fold change ion current (default). For the intra-normalization method, mean and median ion currents are computed by dividing the intensity of each m/z-value of a spectrum by the mean or median of the intensities of all m/z-value of this spectrum. The median fold change approach divides the intensity of each m/z-value of a spectrum by the median fold change between the intensities of all m/z-values of this spectrum and a reference spectrum. The reference is chosen as the median spectrum over all spectra of the data set. For the inter-normalization method, mean and median ion currents are computed by dividing the intensity of every m/z-value within a data set by the mean or median of the intensities of all m/z-values of the same data set. The median fold change approach divides the intensities of all m/z-values within a data set by the median fold change between the intensities of all m/z-values of the average mass spectrum of this data set and a reference. The reference is chosen as the median spectrum over all data sets [125].

Further details about pyBASIS are available in Veselkov et al., 2018.

3.2.2 Reformatting

To ensure independence from MSI instrument and vendor technologies, the alignment and normalization method uses imzml as an input format. After alignment and normalization, all raw data sets are stored in a single file and all processed data

sets in another file, both in HDF5 format [37]. pyBASIS makes use of the HDF5 file format, which organizes the stored data in a folder-like manner. The data sets are stored together with metadata, such as information about the applied processing methods and parameters.

For non-imaging mass spectrometry it was shown that the HDF5-based mz5 format is superior to the mzML format concerning read and write speed and storage space requirements [133]. Since the imzML format is similar to the mzML format, this superiority in performance should still apply. Unfortunately, there are currently no standards for the structural organization of MSI data in HDF5 format. Since there are no standards to follow, the data sets are reformatted in a way to facilitate programmatic accessibility via the pandas software package [76, 117], which is extensively used for the algorithms, methods and tools of this thesis. Therefore, the processed data sets $\mathcal{I}'_q$ are reformatted into a multi-indexed pandas DataFrame $\mathcal{I}''_q$. These DataFrames have the shape $|\rho| \times Z$. Each row is indexed by a triplet (w, h, data set name) and columns are indexed by m/z-values. Consequently, each row of the DataFrame encodes a single mass spectrum and each column encodes a single vectorized m/z-image. Through this reformatting, only the most necessary information is retained for further downstream analysis. The reformatting procedure also creates ibd and imzML files from processed data sets to maintain compatibility with other software packages and tools. Additionally, the reformatting procedure provides an option to set a lower and upper m/z cutoff. This is not only useful to prune the m/z-range of a data set, but also for comparative analysis where the m/z-ranges of different data sets may differ.

After alignment, normalization and reformatting, the data sets $\mathcal{I}_q$ and $\mathcal{I}'_q$ are no longer required for any analysis in this thesis. For convenience and the sake of readability, we will refer to the aligned, normalized and reformatted data set $\mathcal{I}''$ as $\mathcal{I}$ and maintain the structural form introduced in Section 1.1.1. The various data sets of Chapter 2 are still differentiated with the superscripts B, K, S, U, for barley seed, mouse kidney, human skin and mouse urinary bladder, respectively.

3.2.3 Matrix and Artifacts Detection and Reduction

The accurate reduction of tissue-unspecific m/z-signals is an effective way to improve the overall quality of an MSI data set. There are different approaches to detect and reduce tissue-unspecific m/z-signals. For example, if all molecules of the applied matrix are known, the matrix specific m/z-values can be identified and selectively removed. However, this approach requires a fairly accurate mass drift correction

and prior knowledge. Furthermore, there can be m/z-signals that are not matrix exclusive, which has to be taken into consideration. For artifacts, an approach could be to detect and remove m/z-values that show only scattered high-intensity pixels (hotspots) on their m/z-image. Since hotspots are unlikely to represent real molecular distributions, these m/z-values are likely to be artifacts. However, this approach needs a proper definition of a hotspot and the differentiation between hotspots and highly localized tissue-specific signals can be tough.

In our approach for **m**atrix and **ar**tifact **de**tection and **r**eduction (MArDeR), we classify pixels and their associated mass spectra $\mathcal{I}_{h,w}$, with $(h, w) \in \rho$, into the three categories: (A) sample, (B) matrix or (C) artifacts. Then, the mass spectra of the artifact and matrix classes are used to approximate corresponding mass spectra profiles. Pixels are classified as sample or matrix if they have a physical location inside the sample or matrix area, respectively. Pixels are classified as artifacts if their spectra are highly different from all sample and matrix spectra. These artifact mass spectra are typically dominated by m/z-values whose m/z-images show intensity value distributions that are unlikely to represent a real molecular distribution, such as scattered high-intensity hotspots, specific noise distribution pattern or a ubiquitously distributed signal throughout the whole sample area.

The MArDeR module is based on the non-linear dimension reduction method UMAP and works in three steps:

MArDeR step one The first step computes a two-dimensional embedding for the spectral domain $\mathbb{R}^Z \rightarrow \mathbb{R}^2$. Across many alternatives for dimension reduction, the non-linear dimension reduction techniques t-distributed stochastic neighbor embedding (t-SNE) [64] and Uniform Manifold Approximation and Projection (UMAP) [75] generated the best topological structures to allow a reliable visual exploration and selection of the three classes. However, MArDeR integrates UMAP because of its runtime advantage over t-SNE. This is consistent with similar observations from another work, in which dimensional reduction was analyzed in the context of MSI [111].
If the spectral dimensionality Z exceeds a predefined threshold, a faster dimension reduction method is applied to reduce Z to this threshold. Subsequently, UMAP is applied to compute the two-dimensional embedding. For data sets with many pixels of high dimensionality, this improves the runtime of MArDeR without noticeably impairing the quality of the embedding. Due to their fast runtime, principal component analysis (PCA) [89, 45] and latent semantic analysis (LSA) [24] have proven to be suitable methods for this pre-reduction step. MArDeR currently integrates LSA

for the pre-reduction. We performed a few computation time comparisons with the data sets presented in this thesis, and LSA was slightly faster than PCA in most runs. However, since the number of tested data sets and runs was low and the difference were in the range of seconds, these methods seem to be interchangeable to our current knowledge. For a final selection, a more comprehensive study is required, which should be evaluated in terms of embedding quality and computing time. In summary, MArDeR integrates a combination of UMAP and LSA and threshold of 1000 as default $\mathbb{R}^Z \xrightarrow{\text{LSA}} \mathbb{R}^{1000} \xrightarrow{\text{UMAP}} \mathbb{R}^2$.

MArDeR step two The second step focuses on the detection of matrix and artifacts pixels and their differentiation from sample pixels. For this purpose, MArDeR integrates an interactive and an automatic method. Details for both methods are explained below.

MArDeR step three The third step computes the matrix and artifact profiles and reduces the signal strength of their characteristic m/z-values on the entire data set. The matrix and artifact profiles are computed as the arithmetic mean mass spectrum of all pixels of the respective classes. These profiles are then subtracted from every spectral pixel, with the result having a lower limit of zero to avoid negative values. If no pixels have been classified as matrix or artifact, no corresponding profile of this class is computed and subtracted. Depending on the downstream analysis, matrix and artifact pixels may have a negative effect on the applied algorithms. For such cases, MArDeR offers an option to completely remove these pixels from the data set after subtraction of the corresponding profiles.

Automated MArDeR method The automatic method for matrix detection starts with squaring the embedding while preserving the signs of each value. This increases the distances within the embedding, which can result in improved detection performance and allows the parameters to be chosen more generously. Then, a k-Nearest-Neighbors (kNN) graph and a radius-Neighbors (rNN) graph is constructed and their connected components are computed (CC_{kNN} and CC_{rNN}). This approach is similar to the one proposed in Lux et al., 2016 for sequencing contamination detection. To detect matrix characterizing mass spectra, the number of connected components $|CC|$ as well as their compositions are compared. The method can be described as follows:

1. If $|CC_{kNN}| < 2 \ \wedge \ |CC_{rNN}| < 2$: no data is classified as matrix.
2. If $|CC_{kNN}| = 2 \ \wedge \ |CC_{rNN}| = 2$: return the joint set of pixels from the respective smaller components as matrix pixels.
3. If $|CC_{kNN}| = 2 \ \oplus \ |CC_{rNN}| = 2$: select the graph that satisfies the condition and return the pixels of the smaller component as matrix.
4. If $|CC_{kNN}| > 2 \ \wedge \ |CC_{rNN}| > 2$: compute the joint set of pixels of all components from both graphs, except the respective largest, and return this set as matrix.

A major disadvantage of this automated method is that it is not capable to distinguish between artifacts and matrix. If both are present, the method returns the combination of both as matrix. Furthermore, in most of our experiments, some pixels were classified false positively as matrix, which can result in the suppression of tissue-specific m/z-values. Standard clustering algorithms, such as k-means or DBSCAN, were also tested but performed worse than the presented method. The reason is most likely the complex topologies of the UMAP embeddings. If feasible, we would recommend using the interactive MArDeR method to have full control over this sensitive pre-processing step.

Interactive MArDeR method For the interactive method, a tool was implemented that allows the visual exploration of the embedding and the manual selection of artifact and matrix pixels. An example of the tool is provided in Figure 3.2. The interactive MArDeR tool consists of two main panels. The left panel shows the UMAP embedding, where each dot, presented in **blue**, represents a spectral pixel. The right panel is empty by default but uses a binary visualization to show the positions of selected pixels. To make the selection of pixels and the assignment to classes intuitive, a lasso selection tool is combined with class-assignment buttons.
For an initial exploration, all classes are deactivated. This is indicated by the lasso and all selected pixels being colored **black**. Using the buttons located at the bottom of the tool, selected pixels will be assigned to one of three predefined classes, i.e. region of interest (RoI), artifacts or matrix. To visually distinguish the different assignments, RoI pixels are colored in **orange**, artifact pixels are colored in **yellow** and matrix pixels are colored in **green**. While the selections in the embedding are persistent, the binary pixel position visualization shows only the most recent selection. To undo all classifications a reset button is available. Once the classifications have been completed, this must be actively confirmed using the corresponding button. After confirmation three new data sets are created. These

contain only pixels of the three classes artifacts, matrix and sample. While the data sets for artifact and matrix pixels are based on the class selections made, the data set for sample pixels consists of the remaining pixels. Thereafter, these data sets are used to compute the corresponding profiles (arithmetic mean mass spectra) in step three of the MArDeR method. However, if required, these data sets can also be used to examine the mass spectra of the classified pixels in detail by manual analysis.
The indices of pixels assigned to the RoI class can be exported in a npy file [124]. These indices can be used to filter either the embedding or the original data set to perform further downstream analysis.

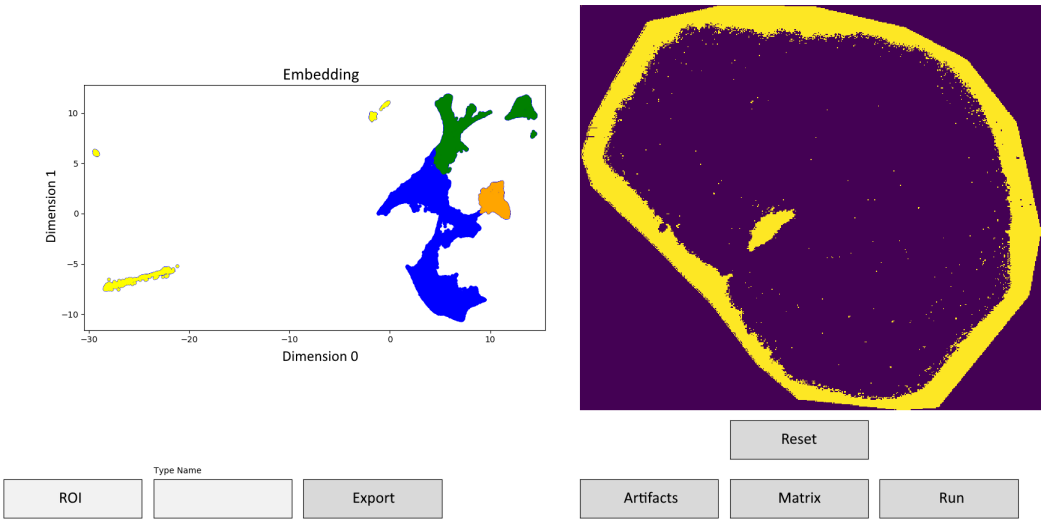


Fig. 3.2.: Exemplary application of the interactive MArDeR tool using the mouse kidney data set $\mathcal{I}^{\mathrm{K}}$ – The left panel shows the LSA-UMAP embedding as described in Section 3.2.3. Each dot represents a spectral pixel, which is initially colored in **blue**. **Orange**, **yellow** and **green** dots represent pixels assigned to the RoI, artifacts and matrix classes, respectively. The right panel shows the binary visualization of the most recent selected, which is the matrix in this case.

Remark If severe artifacts are present, as shown in Figure 3.2, it can be helpful to select and remove these artifacts first and to compute a new embedding without artifacts. This may apply, for example, if the artifacts have a very large distance to other pixels in the high-dimensional space. In the UMAP embedding, this usually results in a large Euclidean distance of artifacts to matrix and sample pixels, which in turn can have a negative effect on the global structure of the embedding, i.e. matrix and sample may be harder to differentiate. By removing the artifact pixels before recomputing the embedding, the topology may be improved, i.e. the differentiation between matrix and sample pixels may be facilitated. If artifacts are present, the benefit of this iterative approach cannot be estimated in advance, at least not to our

knowledge. But since it has no disadvantages apart from the additional computation time, this additional step can still be recommended.

3.2.4 Peak Picking and Deisotoping

For many applications and research questions, only a fraction of the entire measured m/z-range is necessary. Therefore, the selection of a small subset of m/z-values of interest is a commonly applied processing technique, which is referred to as peak picking.

Peak In a theoretical scenario with a perfect spectral resolution a peak would be a single intensity value that measures the relative abundance of a single molecule, i.e. each peak would be associated with one single m/z-value. However, because the spectral resolution is not perfect, a peak appears as a curve. The thinner the curve, the better. In a best-case scenario, a single curve represents the abundance of a single molecule. However, with decreasing spectral resolution the differentiation between peaks can become harder, because problems like overlapping or fusion of curves may appear. Two commonly applied methods to define the relative abundance are either using the intensity value at the peak maximum or using an area under the curve function. Peak picking in ProViM offers both and uses the "full width at half maximum" approach as area function. Since the true m/z-value that corresponds to a peak curve is not known, a representative must be selected. Effective variants are the m/z-value at the curve's maximum or the median or mean of all m/z-values for a peak curve. The peak picking in ProViM uses the median of all m/z-values of the peak curve at full width at half maximum as default and offers the maximum of the peak as an alternative.

Peak picking has several advantages, such as a significant reduction of the file size, which results in reduced computation times, the discarding of background noise, applied algorithms are used only on the m/z-values of interest and exploratory analysis can be performed in a more target-oriented manner. On the other hand, there are also disadvantages, such as selection bias and information discard. In this thesis, three types of peak picking are distinguished: "supervised", "semi-supervised" and "unsupervised".

Supervised peak picking Supervised peak picking refers to the manual selection of a set of precisely defined m/z-values. To use this type of picking, prior knowledge

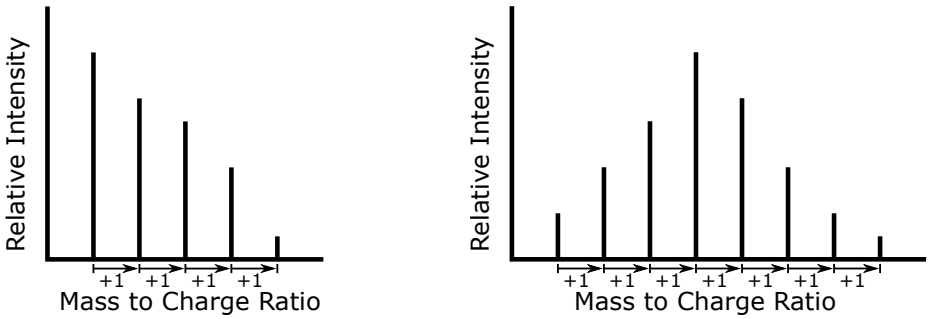
and a defined research question are required. The utilization of an m/z-image browser is a valuable addition to support a precise manual selection of m/z-values. To some degree, such a browser can also compensate for a lack of prior knowledge. This type of picking is not supported in ProViM, because of its unsupervised analysis context, where prior knowledge is often quite limited.

Unsupervised peak picking Unsupervised peak picking refers to the manual or automated selection of a fixed threshold, above which all peaks are selected. This type of peak picking is supported by ProViM.

Semi-supervised peak picking Semi-supervised peak picking refers to the manual selection of a threshold, above which all peaks are selected. Compared to unsupervised peak picking, this variant returns feedback to support a threshold adjustment. Feedback can be provided in various ways and at various levels of detail. ProViM supports semi-supervised peak picking through an interactive tool. Feedback is provided in the form of the number of selected m/z-values and through a visualization of the arithmetic mean mass spectra, where the selected peaks are highlighted.

Isotopes Isotopes are variants of the same chemical element. The different isotopes of a chemical element contain the same amount protons but differ in their amount of neutrons. Consequently, the same chemical element can occur in different atomic masses. If these isotopes are not removed after peak picking, they lead to m/z-image duplicates. This means that several m/z-images show the same molecular distribution with different signal strength because they correspond to the same molecule. From this, it follows that these m/z-images only provide redundant information.

Deisotoping Within a mass spectrum, isotopes for single charged molecules appear as a series of peaks with one Dalton mass difference. There are two isotope patterns that occur very frequently. Both are shown in Figure 3.3. We refer to these patterns as hillside isotope pattern (Figure 3.3A) and hill isotope pattern (Figure 3.3B). The hillside pattern is characterized by a series of peaks that monotonically decrease from a maximum to a minimum. The hill pattern is characterized by a series of peaks that increase to a maximum, followed by a decrease. Due to their consistent structure, these patterns are relatively easy to detect. Although there exist other isotope patterns and this method is a rough approximation, it has the advantage of being very efficient and no prior knowledge is required.



(A) Hillside pattern **(B)** Hill pattern

Fig. 3.3.: Isotope patterns – Two frequently occurring isotope pattern.

ProViM's peak picking module offers unsupervised and semi-supervised peak picking. In both cases, peak detection is performed by detecting local maxima with a predetermined minimum intensity (picking threshold ϕ). Other peak characterizing features, such as shape or area, are ignored. This method has the main advantages that it is highly efficient and it allows an easy backtracing of the results because of the low algorithmic complexity. However, if required, the peak detection procedure could be extended with more complex methods [146, 130, 110, 90]. The deisotoping method is a basic rule-based approximation that uses the hill and hillside isotope patterns, together with a minimum and maximum peak distance (ϕ^- and ϕ^+), to compute potential isotope sets. The use of a range has practical reasons. While both patterns (hill and hillside) have a theoretical distance of exactly one Dalton between each isotope, this is not the case for most practical measurements, due to spectral resolution limitations and measurement inaccuracies. The algorithm for the detection of isotopes can be described as follows:

1. Consecutive peaks within $[\phi^-, \phi^+]$ are computed and collected in isotope sets.

2. Each isotope set is checked, whether it violates one of the two defined patterns. A violation is given for each local minimum of intensity values $\mathcal{I}_{h,w,z} > \mathcal{I}_{h,w,z+1} < \mathcal{I}_{h,w,z+2}$. For every violation, the isotope sets are broken between $\mathcal{I}_{h,w,z+1}$ and $\mathcal{I}_{h,w,z+2}$.

3. Step two is repeated recursively until no set shows a violation.

After the isotope sets have been detected, a representative m/z-value for each set has to be selected. Since the peak with the maximum intensity provides the most intensive m/z-image, this peak is selected as representative by default. Alternatively the representative can be changed to the first peak of the series. After the representative has been selected, all other peaks of the isotope set are discarded.

The unsupervised method requires only a peak picking threshold ϕ and the deisotope range $[\phi^-, \phi^+]$. The method is easy to integrate into fully automated workflows with high throughput but is not suitable for controlled fine-tuning. The semi-supervised approach uses an interactive peak picking tool, which is shown in Figure 3.4. The

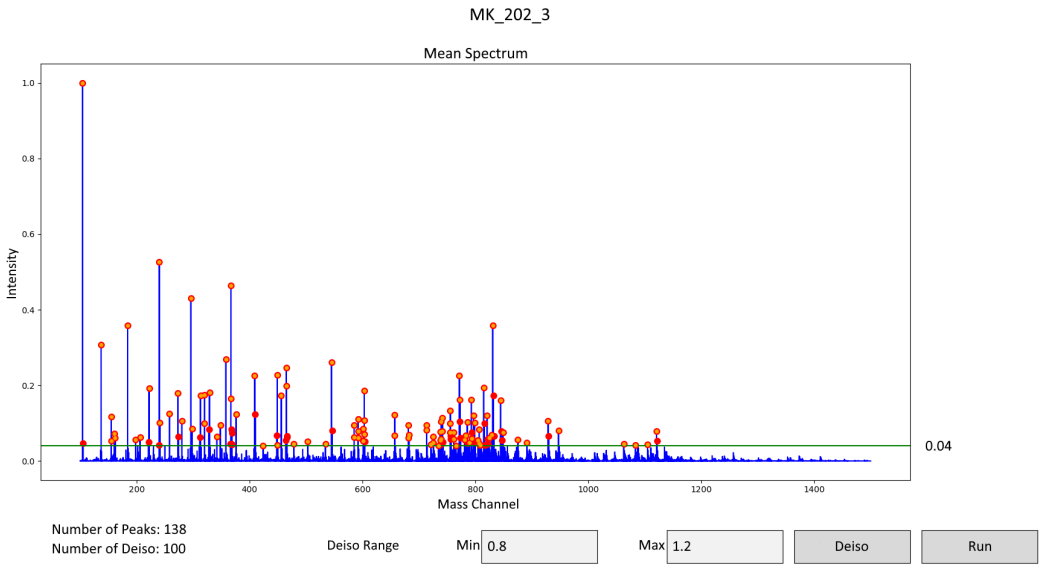


Fig. 3.4.: Interactive peak picking tool – Example of the interactive tool for semi-supervised peak picking on data set $\mathcal{I}^{\mathrm{K}}$. The arithmetic mean mass spectrum is shown in **blue**. The picking threshold is visualized by a **green** line and set to $\phi = 0.04$. The range for deisotoping is set to $\phi^- = 0.8$ and $\phi^+ = 1.2$. The bottom left shows the number of detected peaks (138) and the number of peaks left after deisotoping (100). Detected peaks are marked with **red** circles. The representatives of each isotope set are marked by smaller **orange** circles.

tool shows the arithmetic mean mass spectrum of the whole data set. A **green** horizontal line provides a visual indicator of the current picking threshold ϕ. The threshold changes according to the cursor position. A click interaction (right mouse button) starts the peak detection algorithm. The computed peaks are visualized as **red** circles and the number of computed peaks is displayed at the lower-left corner. After deisotoping, the selected representative peaks for each isotope set are visualized as **orange** circles and the number of remaining peaks is also displayed at the lower-left. Thereafter, the result is passed to the remaining workflow, which creates a picked data set. Deisotoping can be deactivated by setting $\phi^- = 0$ and $\phi^+ = 0$. Because it requires manual intervention the semi-supervised variant is less suited for high throughput workflows. However, especially when no prior knowledge is available, this variant is a valuable tool for data exploration and allows a controlled fine-tuning.

To stabilize the variances of the intensity values and to reduce the effect of outliers, the picked data sets can be transformed by either a square-root transform or a natural logarithm transform. Applying such transformations after peak picking bypasses one of their main disadvantages, which is the increase of noise signals. When processing multiple data sets, it is possible to pick peaks either on the individual arithmetic mean mass spectrum of each data set or on the arithmetic mean mass spectrum of all data sets.

3.2.5 Further Modules

The spatial domain of MSI data sets can contain some offsets or a separate region for matrix measurement. ProViM offers an additional module to remove such offsets and separate matrix regions and to adjust the encoding of the pixel positions accordingly. After peak picking this module was applied to the three data sets $\mathcal{I}^{\mathrm{K}}$, $\mathcal{I}^{\mathrm{S}}$ and $\mathcal{I}^{\mathrm{U}}$.

3.2.6 Application on Real Data

After individual alignment and normalization, the interactive MArDeR method was used independently for the three data sets $\mathcal{I}^{\mathrm{K}}$, $\mathcal{I}^{\mathrm{S}}$ and $\mathcal{I}^{\mathrm{U}}$, followed by peak picking and deisotoping.

In the following, the mouse kidney data set $\mathcal{I}^{\mathrm{K}}$ is used to provide an application example of the whole MArDeR method. $\mathcal{I}^{\mathrm{K}}$ was selected as an example because it exhibits severe artifacts and a matrix that is difficult to separate from the sample, which allows presenting every facet of the method.

Figure 3.5 shows the LSA-UMAP embeddings of $\mathcal{I}^{\mathrm{K}}$ before (Figure 3.5A,B) and after (Figure 3.5C,D) artifact pixel removal. The artifact pixels (yellow) show a clear separation in the embedding. After interactive classification, the artifact profile was subtracted from $\mathcal{I}^{\mathrm{K}}$ and artifact pixels were removed. A new LSA-UMAP embedding was computed for the artifact-free data set to achieve a better separation between matrix and sample pixels. However, Figure 3.6 shows that the classification of matrix pixels is still fuzzy. The embedding shows two clearly separated groups of pixels. Comparing the associated binary visualizations with the brightfield image in Figure 2.3 supports the assumption that these pixels qualify as matrix pixels. Figure 3.6 also shows that by adding pixels from one of the branching regions of the larger pixel group, the binary visualization achieves a more comprehensive coverage of the matrix, including the hole in the tissue that is visible

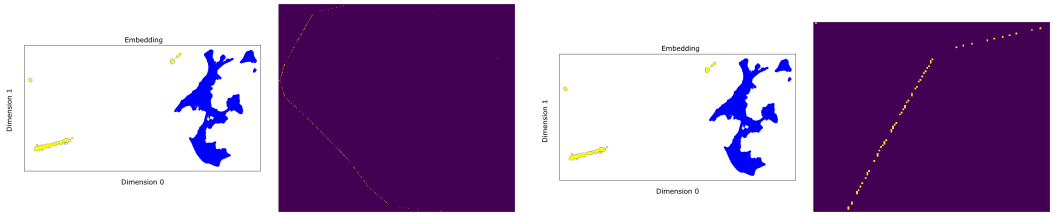
in the brightfield image. The observation that all of the three groups classify as matrix is also supported by Figure 3.7, which shows a comparison of the matrix profiles of the three matrix groups. It can be seen that the overall magnitude of the relative intensity decrease from Figure 3.7A to Figure 3.7C. However, this is to be expected, since the increasingly expanding topology of the three groups in the embedding indicates an increasing variance between the individual mass spectra. Also, some of the mass spectra from the profile of Figure 3.7C will come from the border of the tissue, which increases the variance of intensity values for this profile even more. Although the three matrix profiles show some differences, a general similarity is visible. The similarity is also visible in Figure 3.7D, where all three profiles are superimposed.

Figure 3.6 shows that a large proportion of matrix pixels is closely connected to some of the sample pixels within the embedding. This indicates a strong similarity in molecular composition for some matrix pixels with some sample pixels located at the sample border. We decided not to include the entire branch of pixels visible in Figure 3.6 in the matrix class, but only the part after which the branch spreads out a little. In doing so, we aimed to achieve a balance between the inclusion of enough matrix pixels to create a representative matrix profile and the contamination by too many sample pixels.

After matrix classification, the third step of MArDeR was used to subtract the artifact profile of the first interactive selection and the matrix profile of the second interactive selection from every mass spectrum of the source data set, i.e. the resulting data set after reformatting. To avoid holes within m/z-images, artifact and matrix pixels were not removed. This also includes artifact pixels that were temporary removed to achieve the artifact-free data set for the second embedding computation.

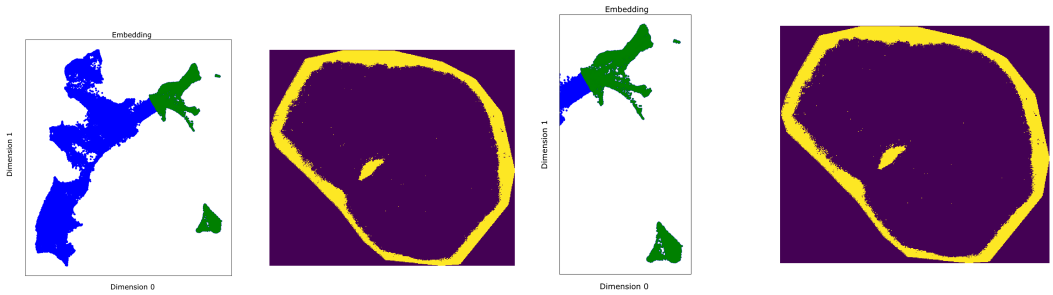
Due to the subtraction, the intensities of the artifact and matrix characteristic m/z-values are not completely removed, but they become considerably smaller (see Figure 3.8). As a result, their influence on downstream analysis is negligible.

The interactive MArDeR method revealed that 6675 pixels were assigned to the pixel position $\mathcal{I}^{\mathrm{K}}_{0,0}$. This is especially interesting because $(0,0) \notin \rho$, which indicates that there may have been an instrument software issue. A total of 6577 artifacts were found, 221 of which were not at position $(0,0)$. This means that 319 of these erroneous mass spectra were not detected through the interactive classification. We assume a connection between the artificial pixels at position $(0,0)$ and the laser movement during the MALDI measurement. The laser moves horizontally across the target. But instead of moving in serpentine lines, at the end of each row, it resets its position by driving back to the beginning of the next row. During this movement, the



(A) Artifact selection

(B) Artifact selection with zoomed binary visualization



(C) Matrix selection

(D) Matrix selection with zoomed embedding

Fig. 3.5.: Interactive artifact and matrix and selection on the mouse kidney data set $\mathcal{I}^{\mathrm{K}}$ – The zoomed binary visualization of the selected artifact pixels (B) reveals artifact signals at the upper left corner of the image ($\mathcal{I}^{\mathrm{K}}_{0,0}$). The zoomed embedding (D) shows that some matrix pixels have a strong overlap to the outer border of the sample, which makes the classification of matrix pixels harder.

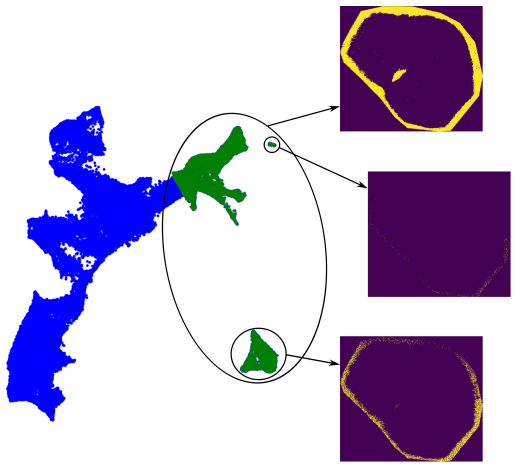
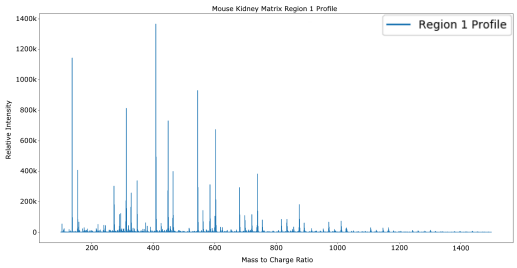
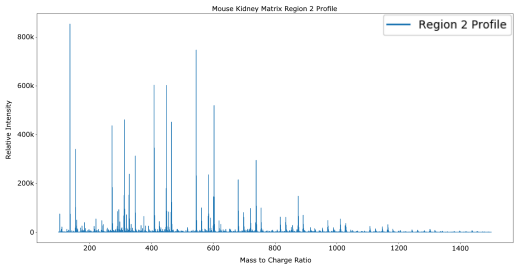


Fig. 3.6.: Matrix pixel comparison – Binary images to compare three different subsets of potential matrix pixels.

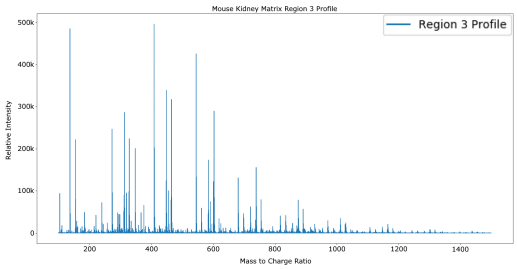
measurement process is not interrupted. We assume that these pixels are the result of measurements that were conducted while the laser resets its position. However, regardless of the correctness of this theory, they are the result of some error, since these pixels are not part of the measured sample area. Therefore, we decided to remove them from the data set. To avoid the analysis of the MArDeR method being



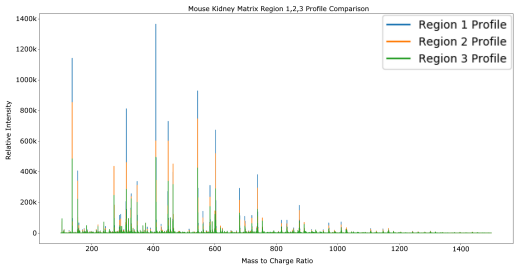
(A) Profile of the small group of matrix pixels



(B) Profile of the medium group of matrix pixels



(C) Profile of the large (branching) group of matrix pixels

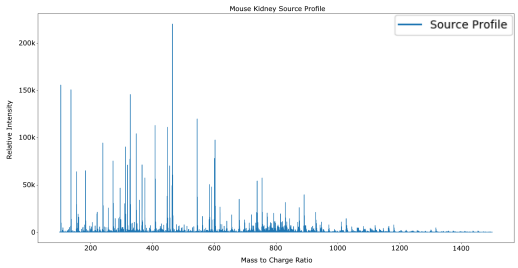


(D) Superimposition of (A),(B) and (C)

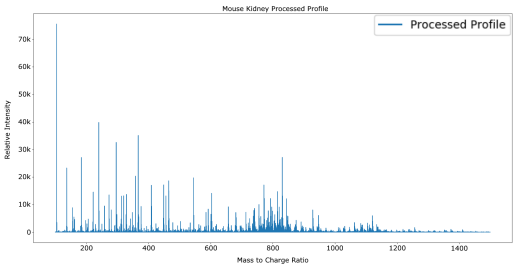
Fig. 3.7.: Matrix profile comparison of the different subsets – (A), (B) and (C) show the matrix profiles of the small, medium and large pixel subset of Figure 3.6. It can be seen that the profiles show some differences in their spectral pattern and overall magnitude. However, the direct comparison in (D) shows that all regions share a similar basic profile. Profile calculation was done after the removal of all mass spectra located at $\mathcal{I}^{\mathrm{K}}_{0,0}$. (An enlarged version of the figures can be found in Figure A.1, Appendix.)

influenced by this particular instrument error, the mass spectra at position $(0, 0)$ are already removed in the following. Therefore artifacts refer to those mass spectra that have been classified as artifacts, except those at position $(0, 0)$.

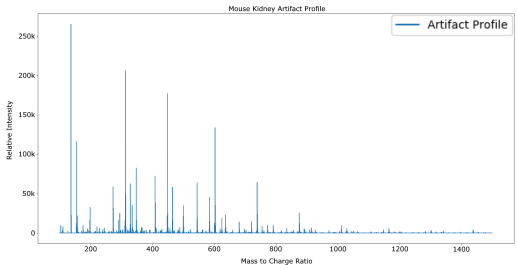
Figure 3.8 shows the arithmetic mean mass spectra of $\mathcal{I}^{\mathrm{K}}$ before artifact and matrix reduction (A), after artifact and matrix reduction (B), the average artifact profile (C) and the average matrix profile (D). It can be seen that all four profiles show major differences. A comparison between the artifact and matrix profile reveals an overlap between some of the most intense peaks. However, there are major differences in the ranking of intensity values. It can also be seen, that the artifact profile shows a suspicious step pattern. Such a regular profile pattern is unusual for biological samples, which further reinforces the suspicion that the corresponding mass spectra are artifacts. A comparison between the unprocessed (A) and the processed (B) profile reveals that the intensities between $\sim m/z$ 150 and $\sim m/z$ 600 are reduced, which in turn results in a relative increase of the m/z area from $\sim m/z$ 700 downstream, especially within the range between $\sim m/z$ 700 and



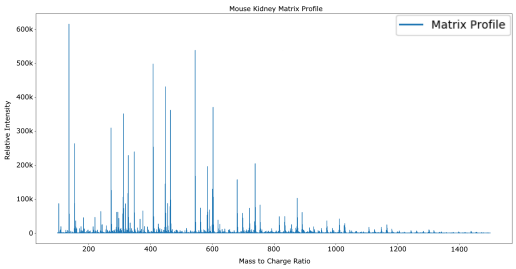
(A) Data set profile spectrum before MArDeR



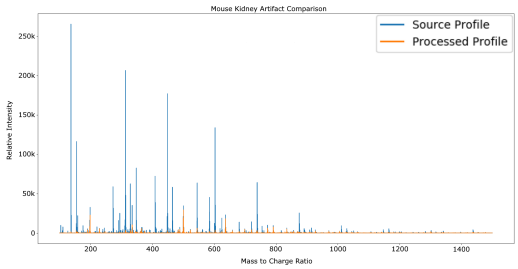
(B) Data set profile spectrum after MArDeR



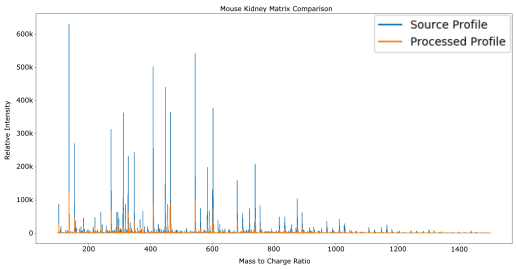
(C) Artifact profile spectrum



(D) Matrix profile spectrum



(E) Artifact profile comparison before (**blue**) and after (**orange**) MArDeR processing



(F) Matrix profile comparison before (**blue**) and after (**orange**) MArDeR processing

Fig. 3.8.: Comparison of spectral profiles – All profiles were computed after the removal of all mass spectra located at $\mathcal{I}^{\mathrm{K}}_{0,0}$. (An enlarged version of the figures can be found in Figure A.2, Appendix.)

$\sim m/z$ 900. Figure 3.8E,F show direct comparisons of the artifact and matrix profiles before and after processing with MArDeR. The comparison reveals that both profiles are considerably reduced.

Matrix signals tend to show signals with high intensities in the mass spectra, which can have a negative influence on the peak picking procedure and downstream visualizations. We have shown both in Wüllems et al., in revision. A good example is the widely used standard matrix DHB, which was also applied for all data sets used in this thesis. DHB produces strong tissue-unspecific signals. A demonstration of the influence of the matrix and artifact reduction on peak picking is presented in Figure 3.9. The figure shows a side-by-side comparison of peak picking

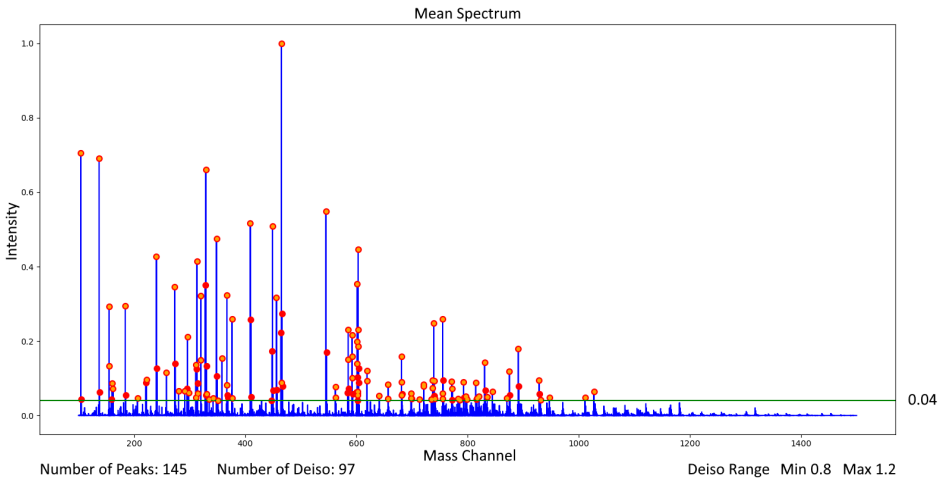
on the profiles of $\mathcal{I}^{\mathrm{K}}$ before and after application of the MArDeR method. The peak picking was computed with the same set of parameters (picking threshold $= 0.04,$ lower deisotope distance $= 0.8,$ upper deisotope distance $= 1.2$). The numbers of detected peaks before MArDeR are 145 without and 97 with deisotoping. After MArDeR these numbers change to 138 without and 100 with deisotoping. The final number of peaks after matrix reduction is slightly higher, which is an early indication that outlier peaks of high relative intensity originating from the matrix suppress tissue-specific peaks. This is in accordance with our previous work [143]. A comparison of the detected m/z-values (peaks) shows that after picking and deisotoping on both profiles, i.e. before and after the MArDeR application, unique subsets of 37 and 40 m/z-values are obtained. Visual examination of these subsets revealed that the m/z-images, which are are unique to the post-MArDeR profile, are much more diverse, i.e. these m/z-images show a wider range of different intensity distribution patterns. Together with our previous work[143], the present example illustrates that the reduction of intensity values of m/z-values, which are characteristic for matrix and artifact, can have a crucial influence on downstream processing, analysis and visualization methods.

Interactive vs automated MArDeR Figure 3.10 illustrates a comparison between the interactive and the automated MArDeR method on the mouse kidney data set $\mathcal{I}^{\mathrm{K}}$. The interactive method is shown in Figure 3.10A,B.

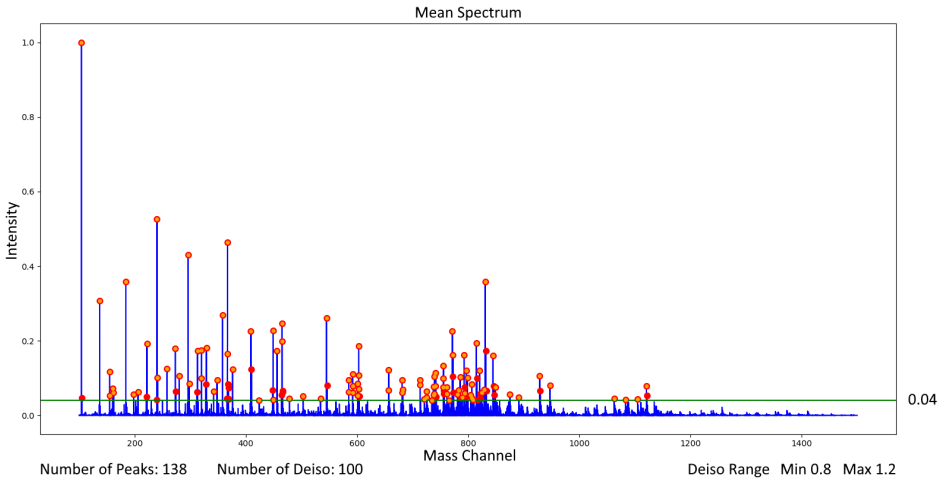
Pixels classified as sample in the first round of colored in **yellow**.

Figure 3.10C,D show the respective results for the automated MArDeR method, with the dots for the remaining pixels colored in **red**.

A direct comparison of both results is presented in Figure 3.10 E,F. As mentioned before, the automated method is not capable to distinguish between matrix and artifacts. Consequently, Figure 3.10 D could not be computed based on the result of Figure 3.10 C. However, to show a comparison between the interactive and automated MArDeR method on the artifact-free embedding, Figure 3.10 D was computed based on the result of Figure 3.10 A. for this particular example, the automated method manages to classify the union of all smaller isolated subgroups as matrix. However, due to the combined nearest-neighbor and radius-neighbor approach, this is only possible if the matrix characteristic mass spectra within the embedding are separated far enough. This is illustrated by the fact that the automated method cannot classify the branching set of pixels as matrix, which is shown in Figure 3.10 F. Figure 3.10 illustrates the two main disadvantages of the automated procedure. First, artifacts cannot be detected and second, if the



(A) Peak Picking on the $\mathcal{I}^{\mathrm{K}}$ profile before MArDeR application

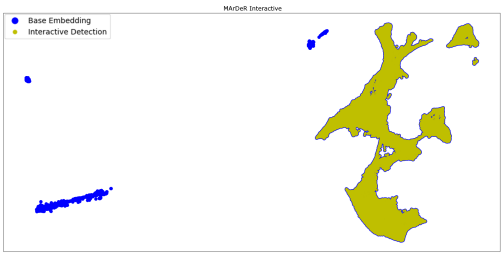


(B) Peak Picking on the $\mathcal{I}^{\mathrm{K}}$ profile after MArDeR application

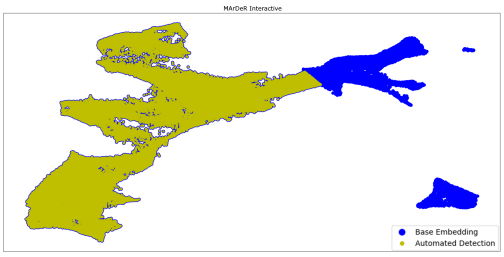
Fig. 3.9.: Comparison of picked m/z-values on $\mathcal{I}^{\mathrm{K}}$ before and after MArDeR application – Picking parameters were: threshold $= 0.04$, deisotope range $= 0.8$ to 1.2. Both profiles show considerably different characteristics, including a different number of selected peaks before (145 for (A) and 138 for (B)) and after (97 for (A) and 100 for (B)) the application of deisotoping.

matrix characteristic mass spectra are not separated far enough from the sample characteristic mass spectra in the embedding, e.g. due to similarity in molecular composition, they cannot be detected.

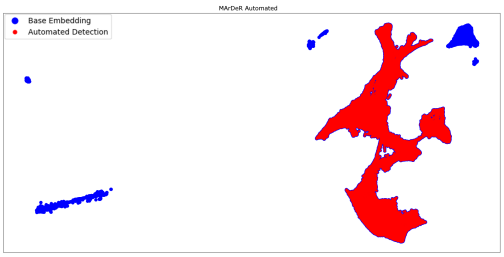
That an automated classification can cause major problems is further illustrated in Figure 3.11, which shows the application of the interactive MArDeR method for the human skin $\mathcal{I}^{\mathrm{S}}$ and mouse urinary bladder $\mathcal{I}^{\mathrm{U}}$ data sets. For both data sets the matrix characteristic mass spectra are hard to detect. For $\mathcal{I}^{\mathrm{S}}$ this could be caused by the molecular complexity of the tissue. For $\mathcal{I}^{\mathrm{U}}$ this could be caused by two factors.



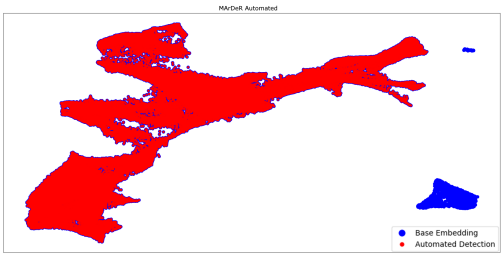
(A) Interactive MArDeR artifact classification



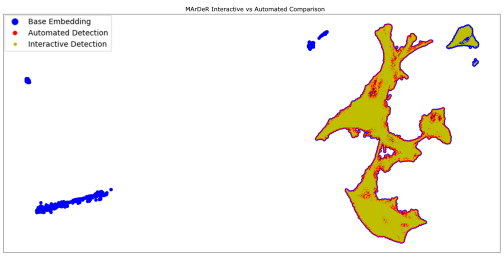
(B) Interactive MArDeR matrix classification on the artifact-free embedding



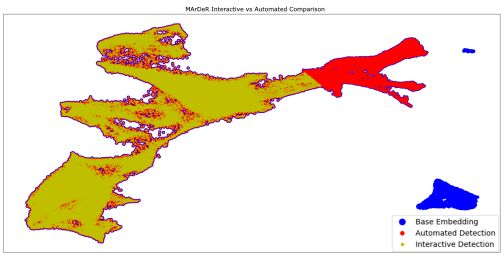
(C) Automated MArDeR matrix classification



(D) Automated MArDeR matrix classification on the artifact-free embedding



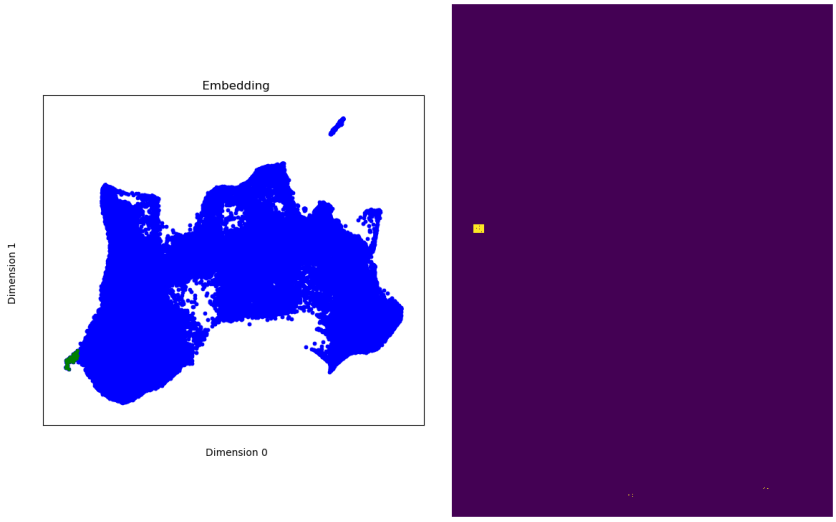
(E) Comparison of automated and interactive MArDeR classifications



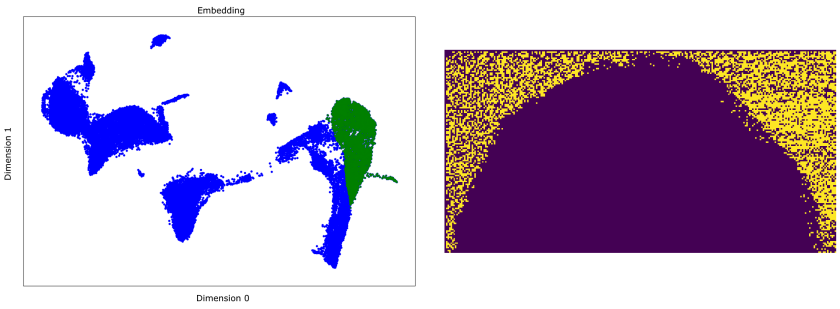
(F) Comparison of automated and interactive MArDeR classifications on the artifact-free embedding

Fig. 3.10.: Comparison of the interactive and automated MArDeR classification results – **Blue** dots represent classification results made, while **red** and **yellow** dots represent the pixels that are classified as sample after applying the automated or interactive MArDeR method, respectively. The artifact-free embedding of (D) follows from the artifact classification in (A).

First, the image scan in fig. 2.7 shows that the tissue has several cuts, which can lead to a strong mixture between sample and matrix characteristic molecules. Second, an in-depth analysis of the matrix showed that this data set contains two different matrix patterns associated with the left and right halves of the tissue. Both factors can complicate the detection of matrix through dimensional reduction. The second factor could also indicate an artificial error factor.



(A) Human skin



(B) Mouse urinary bladder

Fig. 3.11.: Interactive MArDeR for the human skin and mouse urinary bladder data sets – Matrix classification results.

Tab. 3.1.: Summary of the most important processing parameters and results for each data set. – $\mathcal{I}^{\mathrm{K}}$, $\mathcal{I}^{\mathrm{S}}$ and $\mathcal{I}^{\mathrm{U}}$ were processed with ProViM. $\mathcal{I}^{\mathrm{B}}$ was received in a processed state. Z spectral resolution; $|p|$: total number of pixels; $|\rho|$: total number of spectral pixels; ϕ: peak picking threshold; ϕ^-, ϕ^+: lower and upper limit of the deisotope range; Z': remaining number of mass channels after picking; $|\rho'|$: number of selected matrix pixels; $|\rho''|$: number of selected artifact pixels.

Dataset	Z	$\lvert p\rvert$	$\lvert\rho\rvert$	ϕ	ϕ^-, ϕ^+	Z'	$\lvert\rho'\rvert$	$\lvert\rho''\rvert$
Barley Seed ($\mathcal{I}^{\mathrm{B}}$)	N/A	5772	3422	N/A	N/A	101	N/A	N/A
Mouse Kidney ($\mathcal{I}^{\mathrm{K}}$)	70000	205700	147615	0.04	0.8 , 1.2	100	28510	6577
Human Skin ($\mathcal{I}^{\mathrm{S}}$)	20000	110088	75042	0.052	0.8 , 1.2	51	172	0
Mouse Urinary Bladder ($\mathcal{I}^{\mathrm{U}}$)	8562	34840	34840	0.0197	0.8 , 1.2	150	6434	0

Summary Table 3.1 provides a summary of the most important processing parameters and results for each data set used in this thesis.

3.3 Interactive Visual Exploration of Dimension Reduction in Mass Spectrometry Imaging

Dimension reduction is a powerful tool to visually explore and analyze MSI data sets. Two application cases were already presented: the early visual exploration of spatial and spectral features and the detection of matrix and artifact characteristic mass spectra. In addition, the potential to detect and define regions of interest has already been briefly mentioned and will be further discussed below.

To make dimension reduction of the spectral MSI domain easily accessible and to allow a flexible interactive exploration, we have developed the visual analysis (web)tool VAIDRA (**v**isual **a**nalysis tool for the **i**nteractive exploration of **d**imension **r**eduction embeddings of multivariate imaging data and embedding based **a**nnotation). The implementation was done in cooperation with Lillith Bitter as part of her student project. Under my supervision, she implemented parts of the frontend and the backend. Conception, development and design as well as parts of the implementation in frontend and backend were done by me. Following our design guidelines explained in Section 1.4, VAIDRA is kept very simplistic. The whole user interface is presented in Figure 3.12.

VAIDRA provides the functionality to either compute a new embedding or to load a pre-computed one. Both options are easily accessible through a navigation drawer (Figure 3.12a). The user can select a set out of twelve different dimension reduction methods. Since the optimal number of dimensions to explore a data set is usually not known, the number of dimensions can be freely selected.

VAIDRA provides two main visualization domains:

1. The computed sub-embedding of the spectral domain, which is presented as a scatterplot (Figure 3.12c), where each dot represents the projection of an embedded spectral pixel on the selected embedding axes.

2. Colormapped intensity images, including the original m/z-images (Figure 3.12d), the m/z-embedding-images (Figure 3.12e,f) and the RGB projection image (Figure 3.12e).

The simultaneous presentation of an m/z-image and the selected m/z-embedding-images allows an easily accessible comparison between the dimension reduction results and the original molecular distributions. Furthermore, the RGB projection image is integrated, since it is a well-known method to investigate the relationship between different m/z-embedding-images.

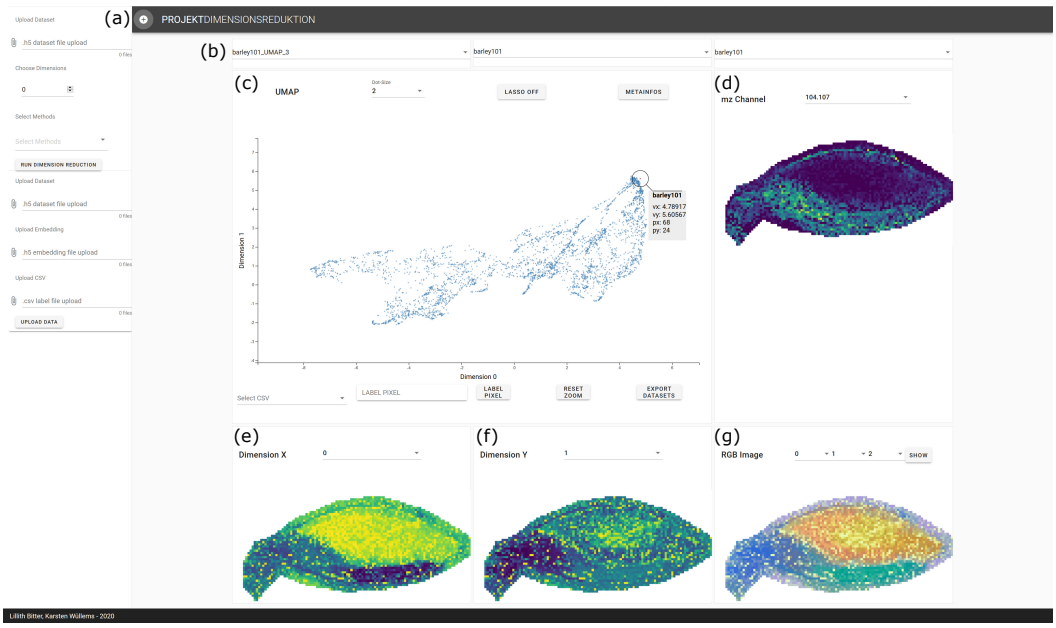


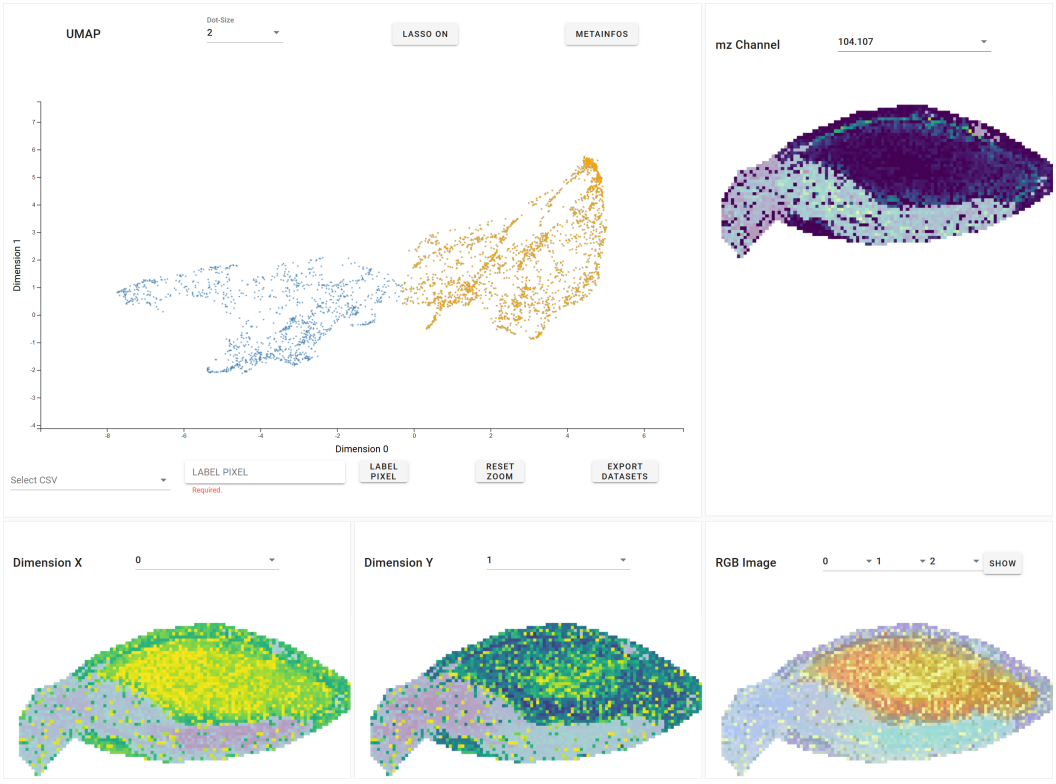
Fig. 3.12.: Overview of the VAIDRA user interface – (a) A sidebar that allows to compute or load a dimension reduction embedding and to load annotation files; (b) selection fields to switch between pre-computed embeddings and to filter by data sets; (c) scatterplot for the sub-embedding of two selected embedding axes; (d) colormapped m/z-image;(e,f) colormapped m/z-embedding-images; (g) RGB projection image.

Interactions To allow an intuitive exploration, the scatterplot provides the basic interaction mechanics zoom, drag, hover and resizing of dots. The visual emphasis of individual data points through the hover interaction is double encoded. The encoding uses **red** color and a surrounding circle that connects the hovered data points to an information panel. The information panel provides basic information about the data set affiliation, the values within the selected sub-embedding and the pixel position in the spatial domain (h, w). The embedding axes to adapt the sub-embedding for the scatterplot visualization can be changed dynamically, which will also adjust the corresponding m/z-embedding-images. An interactive and responsive exploration of both visualization domains is enabled through a lasso selection tool. The lasso selection connects the domains through a link and brush technique. If the lasso selection is used on the scatterplot, the selected data points are colored in **orange** and the associated pixels in all four intensity images are highlighted by reducing the alpha channel of all non-selected pixels. If the lasso is used on one of the intensity images, it will remain visible and the associated data points in the scatterplot will be colored in **orange**. Both lasso select interactions are illustrated in Figure 3.13, using the barley seed data set $\mathcal{I}^{\mathrm{B}}$ and a three-dimensional UMAP embedding.

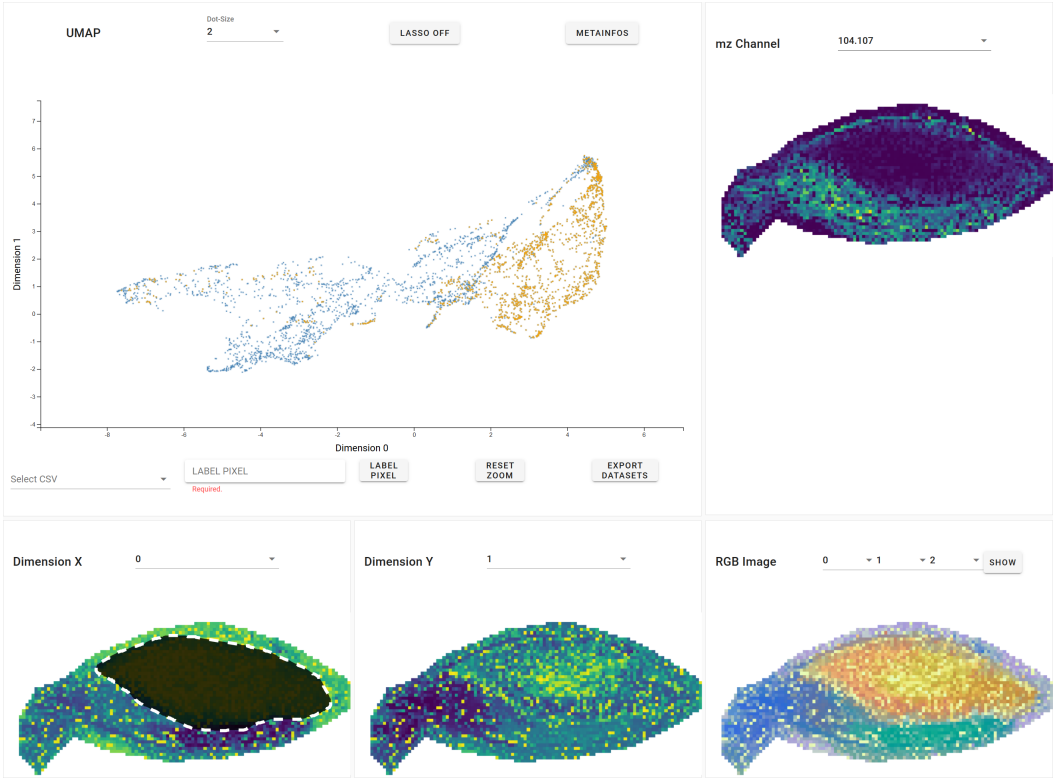
The exploration of different sub-embeddings with VAIDRA can lead to the identification of interesting regions, for example, due to cluster formations in the scatterplot. For such cases, VAIDRA offers an annotation functionality, which is linked with the lasso selection. After a set of pixels has been selected, the selection can be annotated and exported as CSV or HDF5 data set. The CSV stores the pixel positions (h, w) and the label of the annotation. The HDF5 data set is structured as explained in Section 3.2.2. The data set export provides a very flexible alternative to create matrix specific data sets, which could be used for the MArDeR module of ProViM.

To explore how well different samples separate and whether they contain common features, embeddings can also be computed on multiple data sets. Two problems occur if an embedding is computed for several samples. Pixels of different samples must be visually distinguishable within the scatterplot and it is not possible to present the intensity images of all samples simultaneously without losing overview and clarity because usually they are differently shaped. The first problem is solved by encoding the pixels of each sample with different colors. The second problem is solved by using a filter approach, which allows one to select one active data set at a time (see Figure 3.12b). Additionally, a similar filter approach is used to filter the different data sets shown in the scatterplot.

RGB Projection Image An RGB projection image is created by using each of the three color channels (red, green, blue) to encode one image. The RGB projection is a frequently used visualization technique for the analysis of MSI data [32, 101, 93]. It can be used to visualize the interplay of three selected m/z-images to explore co-localization or characteristic spatial features and it can be interpreted as some kind of soft-segmentation map. As shown in Fonville et al., 2013 it can also be used to visualize the relationship of three selected m/z-embedding-images. Because the field of MSI research is familiar with the RGB projection method, VAIDRA provides the ability to create dynamically adjustable RGB projection images (see Figure 3.12g).



(A) Lasso selection, executed on the scatterplot.



(B) Lasso selection, executed on one of the m/z-embedding-images

Fig. 3.13.: Illustrations of the two-way lasso based link and brush interaction in VAIDRA – The lasso selection can be executed on the scatterplot, which highlights pixels in the intensity image domain through the alpha channel, or on the intensity image, which highlights data points in the scatter plot with color.

Application on real data Figure 3.14 presents a small example application of VAIDRA on a three-dimensional UMAP embedding of data set $\mathcal{I}^{\mathrm{U}}$. Selecting the first and third embedding axis for a sub-embedding, two well-separated clusters become visible. The lasso selection reveals that the left-sided cluster (Figure 3.14C,E) contains the pixels of the actual tissue area, while the right-sided cluster (Figure 3.14A) contains the pixels of the matrix area. It is also visible that the left cluster can be divided into two sub-clusters. The upper cluster area (Figure 3.14C) can be assigned to the urothelium, while the lower cluster area (Figure 3.14E) can be assigned to the union of the detrusor muscle and the lamina propria. Comparing this result to Figure 3.11B illustrates that the flexibility provided by VAIDRA is a great benefit, compared to a fixed two-dimensional embedding.

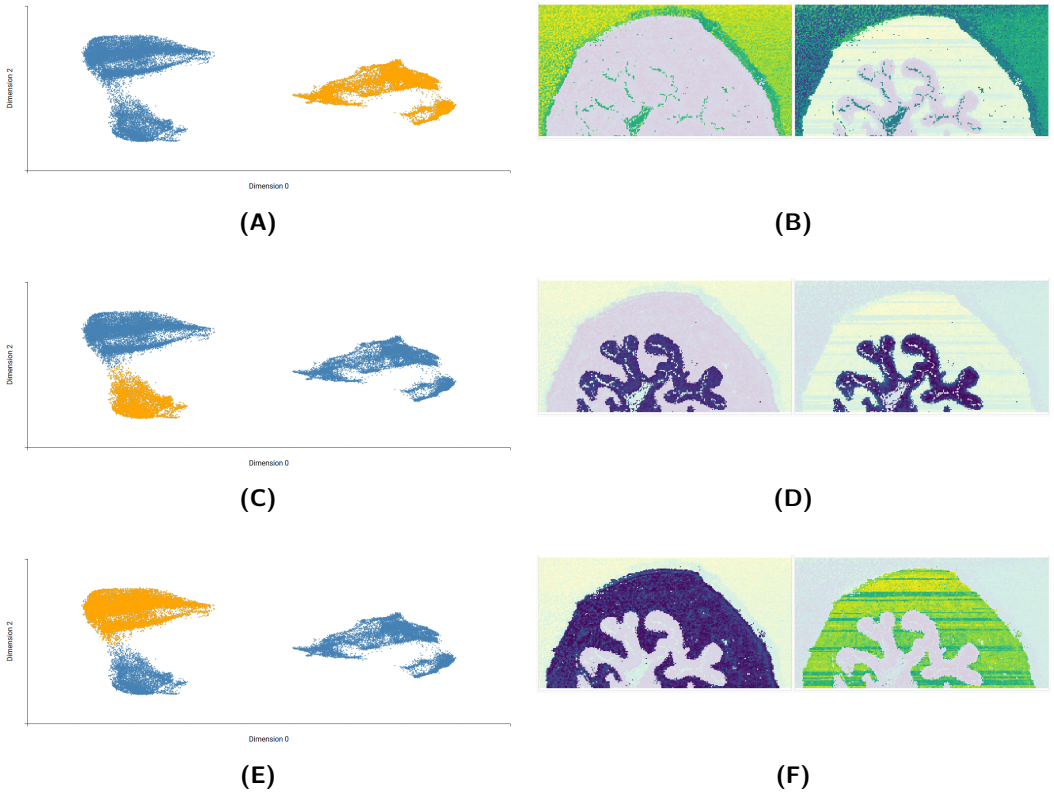


Fig. 3.14.: Example of cluster exploration using VAIDRA – Cluster formations are explored using the lasso selection based link and brush functionality. (A,C,E) show the sub-embedding of the first (x-axis) and third (y-axis) embedding axis of a three-dimensional UMAP embedding on the mouse urinary bladder data set $\mathcal{I}^{\mathrm{U}}$. Lasso-selected data points are colored in **orange**. (B,D,F) show the associated m/z-embedding-images with selected pixels emphasized through opacity (left: projection on the first embedding axis; right: projection on the third embedding axis). The three clusters can be related to (A,B) matrix, (C,D) inner tissue area (urothelium) and (E,F) outer tissue area (detrusor muscle and the lamina propria).

3.4 Summary and Contributions

In this chapter, we presented ProViM, a basic processing pipeline to prepare MSI data for multivariate statistical analysis and visualization. ProViM ensures compatibility with all visual analysis tools presented in this thesis. The pipeline can be used fully automated or interactively, whereby the interactive variant is strongly recommended. The provided interactivity of the modules for detection and reduction of matrix and artifact signals, as well as for peak picking and deisotoping allows full control over these very sensitive processing steps.

We also presented VAIDRA, an intuitive visual analysis tool for a dynamic and interactive exploration of dimension reduction embeddings of MSI data. In addition to its function as an exploration tool, VAIDRA allows the annotation and export of selected pixels as new data set.

To facilitate the execution and ensure the independence of the operating system, VAIDRA is prepared to be executable in a Docker container.

3.5 Improvements and Future Research

To achieve fine-tuned downstream analysis results, different MSI measurements require different processing approaches. Therefore, the inclusion of more processing methods into ProViM would be beneficial. This applies especially to the areas alignment, normalization, variance stabilization and peak picking. ProViM would also benefit from more processing options for the spatial domain, such as hotspot removal. Furthermore, the automatic detection of matrix and artifact signals is still quite error-prone and should be improved. However, the automatic detection of matrix and artifact signals probably constitutes a large new research topic in itself.

Furthermore, although HDF5 is the superior storage format for MSI data, there are no standards for storage in this format. If the community could commit itself to such a standard, this would be an enormous improvement in terms of consistent data storing and sharing.

Although perfectly suited, VAIDRA is not integrated into the MArDeR module of ProViM. Its integration has the potential to greatly improve the detection of matrix and artifact pixels, as it allows a flexible exploration of multiple sub-embeddings. Additionally, the implementation efficiency of VAIDRA offers possibilities for improvement, because performance problems may arise with larger data sets.

4 Co-Localization Analysis in the Spatial Domain

Parts of this Chapter are based on:

- *My master thesis,*
- ***SoRC – Evaluation of Computational Molecular Co-Localization Analysis in Mass Spectrometry Images,***
- ***Detection and Visualization of Communities in Mass Spectrometry Imaging Data***
- ***COBI-GRINE: A Tool for Visualization and Advanced Evaluation of Communities in Mass Channel Similarity Graphs***

`DOIs:`

- `arXiv:2009.14677`
- `10.1186/s12859-019-2890-6`
- `arXiv:2009.11581`

`Code:`

- `https://github.com/Kawue/sorc/`
- `https://github.com/Kawue/msi-community-detection/`
- `https://github.com/Kawue/grine-v2/`
- `https://github.com/Kawue/roi-prediction/`

> *The saddest aspect of life right now is that science gathers knowledge faster than society gathers wisdom.*
>
> **— Isaac Asimov**
> (Science fiction writer and biochemist)

4.1 Motivation

Spatial co-localization analysis of m/z-images focuses on the study of distributions of molecules within biological samples. For this chapter the following assumption is considered to be valid:

Many biological pathways are highly spatially localized within their biological domain.

The given assumption justifies the claim that functionally related molecules are often spatially connected and consequently feature a similar spatial distribution within the sample. Conversely, it can be argued that the co-localization of molecules indicate potential biochemical interactions, which can hint at a relationship within molecular pathways.

m/z-images are descriptions of the distribution of molecules. Thus, co-localization based clustering of m/z-images is an effective way to detect groups of co-localized molecules. If the initially stated assumption holds, then clusters of m/z-images can hint at molecules with functionally linked pathway activity. Thus, the clustering of m/z-images provides a reasonable starting point for pathway-targeted analyses.

However, co-localization based clustering of m/z-images requires functions that quantify the similarity or dissimilarity for two molecular distribution patterns. The definition and selection of such functions are not straightforward. The first part of this chapter will investigate which factors influence the quantification of similarity or dissimilarity between distribution patterns of m/z-images. A particular focus will be on the characteristics of different similarity functions. Furthermore, a workflow scheme will be presented to support the evaluation of different analysis pipelines for the clustering of m/z-images. The purpose of such a workflow scheme is to support users with the design and optimization of their pipelines.

The second part of this chapter will present a new methodological approach to cluster m/z-images. The presented approach is based on community detection, which is a special term for clustering that is widely spread in the social network research community. This new methodological approach requires a mapping of the relationship between m/z-images on a graph structure, which raises an interesting problem. Additionally, an interactive visual analysis tool will be presented to explore the clusters of the graph-mapped m/z-images.

Finally, a method will be proposed to approximate regions of interest based on clusters of co-localized m/z-images.

4.2 Quantification of Co-Localization of Molecular Distributions

Many previous works on MSI data preferably used quite basic image comparison functions to quantify the similarity for a pair of m/z-images, such as *Pearson Correlation Coefficient* [71], *Cosine Similarity* [71] or *Structural Similarity Index* [128]. But most of these works do not provide any motivation for the choice of their function. Recently, Ovchinnikova et al., 2020 presented a small comparison study of some unsupervised similarity functions and supervised learning functions, also including the three above. On their presented gold standard data set the best results were achieved by the application of the *Cosine Similarity* in combination with a basic pre-processing, which consisted of median thresholding and a sliding square window median filter of size three.

The analysis of the spatial MSI domain is mainly focused on the analysis of m/z-images. This leads to a close relation to the field of image analysis. In image analysis, it has already been shown that the degree of similarity between two images cannot be quantified in a universal way. The reason is that visual similarity is often context-dependent. This is supported by previous works. Those demonstrate that for certain imaging domains, such as natural scenes [109, 113, 106] or face images [1], some similarity functions perform significantly better than others.

In image analysis, the context can be divided into two different concepts:

1. Context given by the technical recording conditions.
2. Context given by the image scenery.

For a comprehensive image analysis, both types of context are important. For this reason, the field of image analysis has already proposed a vast amount of functions to quantify the similarity between two images. This raises the question, whether the importance of context also applies to m/z-images, i.e. are there functions that are better suited than others to quantify the similarity between two m/z-images?

Several factors influence the properties of an MSI data set, such as resolution, intensity range, pixel-to-pixel variation or noise. These factors can be differentiated into experimental/technical factors, such as differences in MSI instruments, measurement parameters, matrix application protocols, and biological/biochemical factors. Both factors can be related to the concepts of context described for image analysis. The experimental/technical factors can be compared to the technical recording conditions, while the biology and biochemistry of the sample can be compared to

the image scenery. However, there is a third type of context that must be considered during the exploration, evaluation and interpretation of computer-aided analysis results. This third type of context can be seen as a kind of algorithmic context and refers to the selection of the applied algorithms and their parametrization. This includes in particular algorithms for the pre-processing of the data up to the analysis and post-processing. In application-oriented terms, the algorithmic context describes the setup of the analysis pipeline. Therefore, the term pipeline setup will be used to refer to this kind of context. Different pipeline setups can change the properties of the recorded data set and the properties of the analysis result. This is illustrated in Figure 4.1 using the cluster analysis of m/z-images as an example.

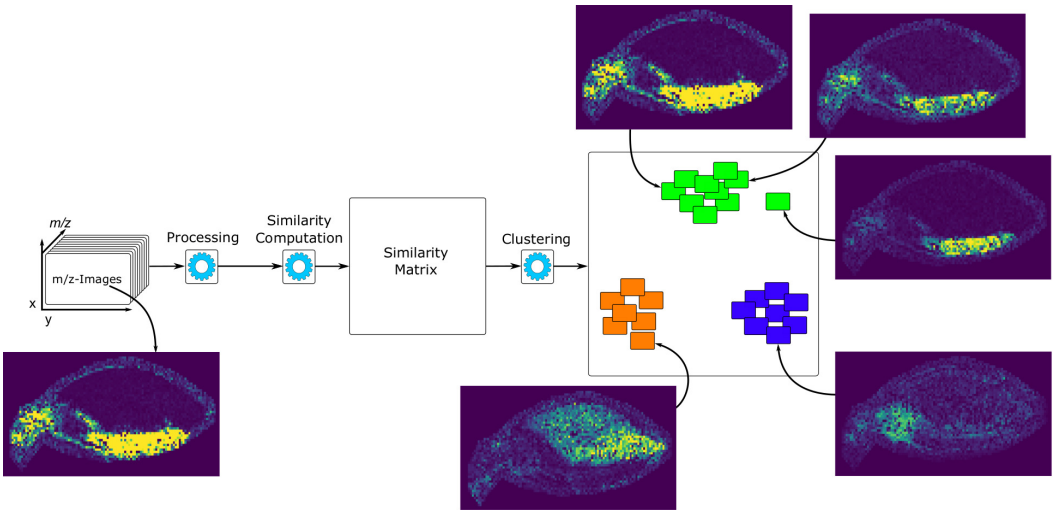


Fig. 4.1.: Illustration of a pipeline for the cluster analysis of m/z-images – The analysis result is influenced by the data source and the configuration of the pipeline setup, i.e. the (pre-)processing, the similarity function and the clustering method.

It is well known from the field of image analysis that the context is relevant for the quantification of the similarity between two images. This leads to the question about the influence of the pipeline setup for the analysis of m/z-images.

In summary, we assume that the co-localization analysis of m/z-images depends on three types of context:

1. Context given by experimental and technical factors.
2. Context given by biological and biochemical factors.
3. Context given by the pipeline setup.

This supports the initial claim that the selection of a similarity function for clustering m/z-images is not straightforward and should be questioned. Furthermore, the

question arises on how strong the influence of sub-optimal pipeline setups is on the result of the analysis.

Remark The notions of similarity and distance are often interchangeable since the results of the corresponding functions can be reversed by applying a compatible mapping function. This means, that although only the notion of similarity is used in this chapter, the presented findings can usually also be extended to the notion of distance. Examples of two commonly used mapping functions to convert a similarity into a distance and vice versa are given in Equations (4.1) and (4.2).

$$\delta(x) = 1 - \hat{\delta}(x) \tag{4.1}$$

$$\delta(x) = \frac{1}{1 + \hat{\delta}(x)} \tag{4.2}$$

where δ defines a similarity function and $\hat{\delta}$ a distance function or vice versa.

Equation (4.1) is usually used if $\hat{\delta}(x) \in [0, 1]$ or $\hat{\delta}(x) \in [-1, 1]$, where a result in the range $[-1, 1]$ can be normalized with a factor of 0.5 to ensure that the mapped result remains in the range $[0, 1]$. Equation (4.2) is usually used if $\hat{\delta}(x)$ has no upper limit. Especially for functions whose results have no lower or upper limits, it might be necessary to use a linear transformation to scale the results into a fixed range. Equation (4.3) can be applied to scale any set of numbers to any range. Equation (4.4) represents a special case of Equation (4.3) and scales any set of numbers to the range $[0, 1]$.

$$f(x_i) = (b - a)\frac{x_i - \min(x)}{\max(x) - \min(x)} + a \tag{4.3}$$

which is a mapping of $[\min(x), \max(x)] \mapsto [a, b]$.

$$f(x_i) = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{4.4}$$

which is a special case of Equation (4.3) with $a = 0$ and $b = 1$.

4.2.1 Image Pattern Regularity

There is no generally accepted terminology for describing the morphology and structure in a biological sample beyond single chosen contexts, such as histopathology classification schemes. The same applies to the spatial domain of MSI data, as there is no generally accepted terminology for the description of intensity patterns in

m/z-images. However, we assume that the structure of the occurring patterns may influence the quantification of the similarity between m/z-images. For this reason, a system for classifying a biological sample and its associated data set could be beneficial.

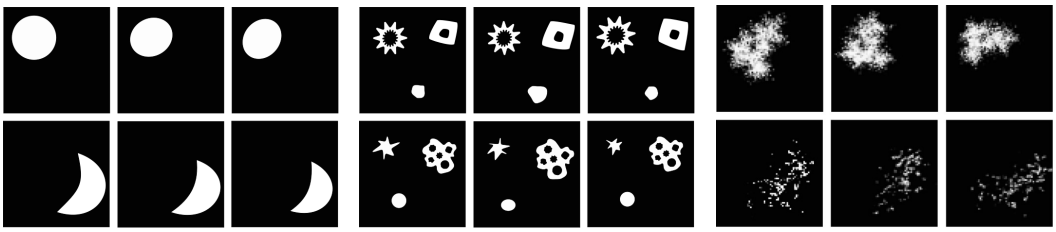
In previous works, morphologies and structures are described with one-dimensional quality scales using terms such as *simple, regular, partly regular, irregular* or *complex* [96, 119, 149, 38, 145]. Following this observation, we also apply such a scale to describe and rank the structures in biological samples. The scale is based on the degree of *regularity* and ranges from *regular* to *irregular*.

To describe the term regularity, the following definition is proposed:

The degree of regularity in a pattern is related to the effort required to describe the pattern.

In application-oriented terms, this means that the degree of regularity is related to the number of points, lines or geometrical shapes necessary to describe the pattern, i.e. a low number indicates a highly regular pattern and a high number indicates a highly irregular pattern.

To give an intuition for the assessment of different levels of regularity using this expression, Figure 4.2 illustrates examples of three different levels of regularity using abstract shapes.



(A) High regularity **(B)** Medium regularity **(C)** Low regularity

Fig. 4.2.: Three artificial image data sets of abstract shapes to exemplify pattern regularity – The three data sets consist of six images each. Images of the upper and lower row for each example are considered to be similar enough to build a cluster. (A) All patterns are areal, crisp, well defined and the pattern of the upper and lower row are well disjunct. (B) The patterns are finer and less areal, but still well defined. The pattern of the upper and lower row do partially overlap, but due to their difference in shape, they are still well disjunct. (C) The signal distribution patterns are noisy, filigree and not well defined. The pattern of the upper and lower row are still disjunct, but not as clearly as in (A).

In the following, the proposed concept is used to categorize the samples and the corresponding data sets of this thesis according to their regularity.

The barley seed $\mathcal{I}^{\mathrm{B}}$ is highly regular. It consists of a small number of compartments, which are disjunct and well defined. The mouse kidney $\mathcal{I}^{\mathrm{K}}$ is also highly regular. It has a well-structured morphology and consists of a small number of compartments. However, it is slightly less regular than the barley seed, since the compartments are less disjunct. The mouse urinary bladder $\mathcal{I}^{\mathrm{U}}$ shows a medium regularity. Most of its morphological structures are well defined and many of the molecular distribution patterns are disjunct. However, some of the morphological structures are filigree, which can also be seen in the m/z-images. Additionally, some of the m/z-images are quite noisy. The human skin $\mathcal{I}^{\mathrm{S}}$ has a more irregular morphology. The molecular distribution patterns are partially blurred and they partially overlap each other. In addition, some of them are wide-spread, while others are quite small and densely structured.
In summary, $\mathcal{I}^{\mathrm{B}}$ shows the highest degree of regularity, followed by $\mathcal{I}^{\mathrm{K}}$, further in the middle of the regularity spectrum follows $\mathcal{I}^{\mathrm{U}}$ and the lowest regularity is shown by $\mathcal{I}^{\mathrm{S}}$.

4.2.2 Mass Channel Image Features and Representations

In order to encode pixel-related mass signal characteristics beyond the measured intensity value, various feature maps and image representations are defined in the following.

Gradient vector image: To encode the local intensity changes and their directions (i.e. the intensity value gradient) within an m/z-image, a gradient vector image $\mathcal{I}_z^{\nabla}$ is computed for each m/z-value. The gradient vector image is defined as follows:

$$\mathcal{I}_z^{\nabla} = \nabla_{h,w}\mathcal{I}_z = \begin{pmatrix} \nabla_h \mathcal{I}_z \\ \nabla_w \mathcal{I}_z \end{pmatrix} \quad (4.5)\ , \qquad \nabla_h \mathcal{I}_z = \frac{\partial \mathcal{I}_z}{\partial h} \quad (4.6)\ , \qquad \nabla_w \mathcal{I}_z = \frac{\partial \mathcal{I}_z}{\partial w} \quad (4.7)$$

$\nabla_h \mathcal{I}_z$ and $\nabla_w \mathcal{I}_z$ are the partial derivatives of $\mathcal{I}_z$ with respect to h and w. Both derivatives describe the local directional changes of intensity values in their respective horizontal or vertical direction.

Magnitude image Based on the gradient vector image the strength of the local directional changes can be quantified. The resulting image is known as the magnitude image ($\mathcal{I}_z^{\mathrm{M}}$) and is defined as follows:

$$\mathcal{I}_z^{\mathrm{M}} = |\mathcal{I}_z^{\nabla}| = \sqrt{(\nabla_h \mathcal{I}_z)^2 + (\nabla_w \mathcal{I}_z)^2} \quad (4.8)$$

Orientation image The orientation image ($\mathcal{I}_z^{\mathrm{G}}$) can be interpreted as a complement to the magnitude image and can also be derived from the gradient vector image. While the magnitude image describes the strength of each local change, the orientation image describes the direction of each local change. The direction is expressed by a number from the range $[0, 360]$. The orientation image is defined as follows:

$$\mathcal{I}_z^{\mathrm{G}} = \frac{180}{\pi}\, tan^{-1}\left[\frac{\nabla_h \mathcal{I}_z}{\nabla_w \mathcal{I}_z}\right] + 180 \tag{4.9}$$

Vectorization The vectorization of an image $f : \mathcal{I}_z \mapsto \mathcal{I}_z^{\mathrm{V}}$ describes the process to transform a two-dimensional image $\mathcal{I}_z \in \mathbb{R}^{(H \times W)}$ into a one-dimensional vector $\mathcal{I}_z^{\mathrm{V}} \in \mathbb{R}^{(|\rho|)}$, which contains only spectral pixels. A common method, which is also applied here, is the "stacking of pixel rows" (also referred to as "row major order"):

$$\mathcal{I}_z^{\mathrm{V}} = \left[\mathcal{I}_{0,0,z}, \ldots, \mathcal{I}_{0,w,z}, \ldots, \mathcal{I}_{h,w,z}, \ldots\right] \tag{4.10}$$

Image histogram Histograms of intensity values are another popular tool for the analysis of images. The one-dimensional histogram of a two-dimensional image $\mathcal{I}_z \in \mathbb{R}^{(H \times W)}$, over all spectral positions, is referred to as $\mathcal{I}_z^{\mathrm{P}}$.

Each of the presented feature maps and image representations is regularly used in the field of image analysis. Therefore, it is reasonable to assume that they can be useful in the context of m/z-image analysis.

4.2.3 Image Scale-Space Representations

The computation of image scale-space representations is a processing method that allows analyzing structures and details of different granularity in images [134, 62]. The motivation behind the scale-space theory is a concept derived from the following real-world observation:

> *Objects can be perceived in different ways depending on the scale of observation. On different scales, they appear to be composed of different structures.* [61]

Lower scale features emphasize finer structures (like edges, corners and hotspots), while higher scales should emphasize coarser structures (like homogeneous regions). Biological entities, such as cells and tissues, are also organized and structured on

several scales. Therefore, the use of scale-space representations for the analysis of m/z-images is well motivated.

The scale-space representation is a family of images generated by repeated convolution with a two-dimensional Gaussian kernel (g), where each repetition defines a single scale level (t). Formally, a scale-space representation of an m/z-image can be defined as:

$$\begin{aligned} g(h, w; t) &= \frac{1}{2\pi t} e^{-\frac{h^2+w^2}{2t}} \\ L(h, w; t) &= g(h, w; t) * \mathcal{I}_z, \text{ with } * \text{ denoting a convolution.} \end{aligned} \tag{4.11}$$

The ";" in g and L indicates that the convolution is performed over the pixels h and w only, while the parameter t specifies the predefined scale level. Two special cases for t are $t = \sigma^2$, which corresponds to the convolution with a standard Gaussian filter and $t = 0$, which corresponds to the identity transformation $L(h, w; 0) = \mathcal{I}_z$. In this thesis Equation (4.11) is adjusted by setting $t = \sigma^2$. In addition, a step size variable $s \in \mathbb{N}^0$ is introduced. This changes the definition of $g(h, w; t)$ to $g(h, w; s\sigma^2)$ and $L(h, w; t)$ to $L(h, w; s\sigma^2)$. Consequently, $L(h, w; s\sigma^2)$ denotes an s-fold convolution of $g(h, w; \sigma^2) * \mathcal{I}_z$.

An example of the application of scale-spaces on m/z-images is given in Section 4.2.8.

4.2.4 Evaluation of Pipeline Setups

To evaluate different pipeline setups for the analysis of m/z-images, we propose a methodology called SoRC (**s**imilarity **o**f **r**anked **c**luster indices). SoRC is a workflow scheme that consists of three parts:

a) Pre-processing, e.g. transformation and enhancement of the data.

b) Analysis, e.g. clustering.

c) Evaluation, i.e. computation of SoRC scores.

Part a) generates a collection of different data sets through the optional application of different pre-processing procedures. A pre-processing procedure can refer to the application of a single method or a series of methods.
Part b) describes the algorithmic framework to achieve the required analytical output. This often refers to the application of data mining or machine learning methods. The combination of pre-processing, algorithmic framework and parameterization

that leads to an effective information gain for a specific data set is usually not known in advance. Therefore, different setups should be applied, evaluated and compared. Part c) consists of the computation and visualization of SoRC scores for each analytical output computed in part b). The visualization is implemented as a bar-chart that is integrated within a table. This tabular visualization will be referred to as bar-chart-table.

As motivated above, the analysis objective for this chapter is to find clusters of m/z-images that show co-localized molecular distributions, i.e. similar distribution patterns of intensity values.
For the specific application in this context, part a) is subdivided into two procedures: (P_1) thresholding and smoothing and (P_2) computation of scale-space representations. P_2 is carried out in two variants, referred to as P_{2a} and P_{2b} (see details below). The two pre-processing procedures are tested individually and in combination, which leads to six different pre-processing setups: 1. no pre-processing, 2. P_1, 3. P_{2a}, 4. P_{2b}, 5. P_1 + P_{2a} and 6. P_1 + P_{2b}.
After pre-processing follows the algorithmic framework, i.e. the application of the actual data analysis method. For clustering of m/z-images the algorithmic framework consists of two parts. First, the quantification of pairwise similarities between m/z-images to create a similarity matrix and second, the clustering. For the analysis setup, part b) combines thirteen similarity functions with three clustering methods. Together with the six different pre-processing setups from part a) this results in $6 \cdot 13 \cdot 3 = 234$ different clustering results for one single data set. However, the scale-space representations are not utilized by every similarity function, which means that the corresponding functions only lead to two different pre-processing setups (no pre-processing and P_1). This reduces the actual number of different results to 138, which is still too much to evaluate manually.
All these clustering results are forwarded to part c), where the SoRC scores are computed and visualized.

The implementation of the SoRC workflow as used in this thesis, with the clustering of m/z-images as analysis objective, is presented in Figure 4.3. Since this section focuses on similarity functions, only two pre-processing methods (combined to six different setups, see above) and three clustering methods are considered, while thirteen similarity functions are considered. In Section 4.2.8, the SoRC workflow as shown in Figure 4.3 is used to present an application on real data.

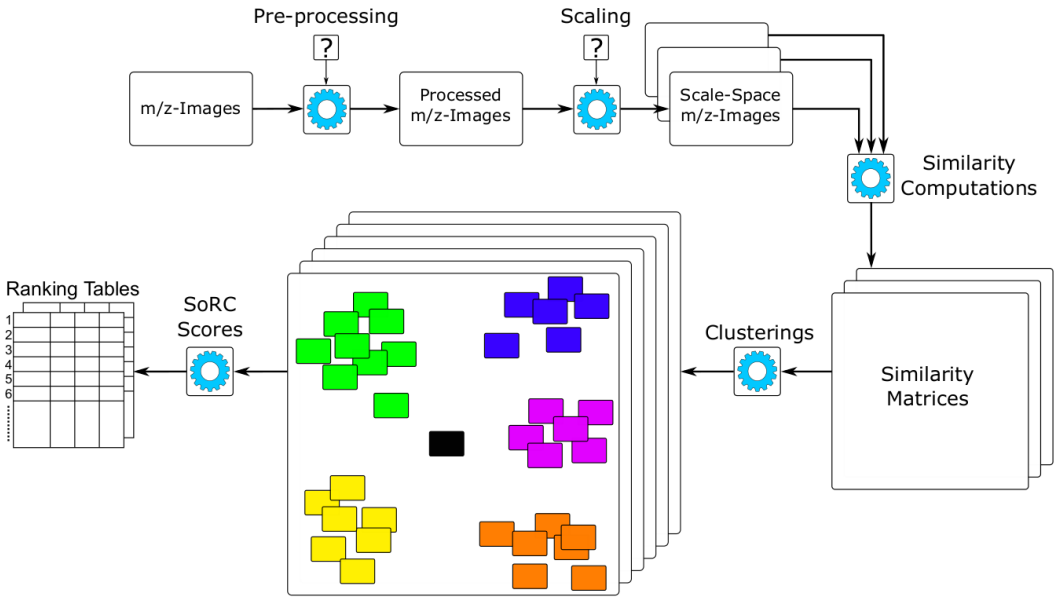


Fig. 4.3.: The SoRC workflow principle for evaluating different pipeline setups for the clustering of m/z-images – The presented SoRC workflow executes several pipeline setups for the clustering of m/z-images. The individual pipeline setups are evaluated using a quality estimation score (SoRC score) and visualized with ranking tables (bar-chart-tables). The pipeline setups are built from optional pre-processing, optional computation of different scale-space representations, the application of several similarity functions to quantify the similarity between every pair of m/z-images for the computation of similarity matrices and the clustering using different methods. The various clustering results (analytical outputs) are evaluated using the SoRC scores and visualized to support the selection of a pipeline setup. The technical/experimental and biological/biochemical context is inherent to the data source (m/z-images).

4.2.5 SoRC Score

The SoRC score is used to estimate the quality of a clustering result and thus the effectivity of the entire pipeline setup with a single number. The computation of a single score allows an easy ranking, visualization and comparison for a multitude of different setups, i.e. different methods, modifications and parameterizations. This reduction step thus aims to increase the efficiency in the process of designing and improving software pipelines for the clustering of m/z-images. A single score also has the advantage to provide an intuition about the quality differences between different pipeline setups. This can be particularly useful if a trade-off between computational resources, i.e. time or power and quality is required.

Clustering quality In general, clustering is considered to be of higher quality if the resulting clusters are dense, well separated and represent the most important struc-

tural features of the data. Furthermore, a high-quality clustering should generate a descriptive model of the data distribution that can be used for new interpretations and insights.
The SoRC score is based on two different cluster indices, which are not strictly correlated, to express these properties in numbers and estimate the clustering quality:

1. The Silhouette Coefficient score [100]
2. The Calinski-Harabasz index [17]

Silhouette Coefficient Score The Silhouette Coefficient is based on the comparison of the similarity of a data point to its assigned cluster (cohesion) and to the most similar other cluster (separation). The Silhouette Coefficient score (SCS $\in [-1, 1]$) is the arithmetic mean over the Silhouette Coefficients of all data points. The higher the number of unambiguous cluster assignments, the higher the SCS value. Thus, SCS $= 1$ indicates high-quality clusters, SCS $= 0$ indicates ambiguous clusters, which is a sign for overlapping clusters structures, and SCS $= -1$ indicates low-quality clusters.

Calinski-Harabasz Index The Calinski-Harabasz index (CHI $\in [0, \infty]$) is defined as the ratio between the within-cluster dispersion and the between-cluster dispersion. The index increases with decreasing intra-cluster variance and increasing inter-cluster distance. Thus, the index is higher when clusters are dense and well separated.

A main difference between these two cluster indices is that the SCS is computed on the actual similarity values, while the CHI is computed directly on the data points within the clusters. So the integration of both indices into the SoRC score enables the evaluation of both aspects, the composition of the similarity values and the composition of the data points (i.e. m/z-images) within each cluster.

If pre-processing is applied, the CHI is calculated based on the pre-processed m/z-images. However, scale-space representations are ignored and the CHI is always computed on the zero-level scale.

Another difference between both indices is the value range of the output values. The SCS is bound between $[-1, 1]$, while the CHI is bound between $[0, \infty]$.

To make both indices comparable, the ranks of the values are considered instead of the actual values. So for both indices, the final values are ranked and the rank position values are normalized into $[0, 1]$. Consequently, higher ranks represent

better clusters. Ties are solved by mean ranks, e.g. the values $[1, 0.7, 0.7, 0.7, 0.4]$ rank to $[5, 3, 3, 3, 1]$. The actual SoRC score is the arithmetic mean of the normalized ranks of both cluster indices.

SoRC score calculation For a given data set $\mathcal{I}$ without any pre-processing and a selected similarity function $\mathcal{F}_n$ out of a set of similarity functions $\mathcal{F}$, the SoRC score ψ is calculated as follows. Let $\mathfrak{C}(\mathcal{F}_n(\mathcal{I}))$ denote the function that returns a clustering result using a similarity function $\mathcal{F}_n \in \mathcal{F}$. Let $Q_{\text{SCS}}(\mathcal{F}_n)$ denote the rank of the SCS value in the set $\{\text{SCS}(\mathfrak{C}(\mathcal{F}_n(\mathcal{I})))|\mathcal{F}_n \in \mathcal{F}\}$ of all SCS values. Let $Q_{\text{CHI}}(\mathcal{F}_n)$ denote the rank of the CHI value in the set $\{\text{CHI}(\mathfrak{C}(\mathcal{F}_n(\mathcal{I})))|\mathcal{F}_n \in \mathcal{F}\}$ of all CHI values. For both sets applies that a larger value is assigned a larger numerical value in the ranking. The SoRC score is then defined by Equation (4.12).

$$\psi(\mathcal{F}_n(\mathcal{I})) = \frac{1}{2|\mathcal{F}|}(Q_{\text{SCS}}(\mathcal{F}_n) + Q_{\text{CHI}}(\mathcal{F}_n)) \tag{4.12}$$

If pre-processing is considered part of the pipeline setup, a single SoRC score is calculated for each combination of pre-processed data set variant $\mathcal{P}_m \in \mathcal{P}$ and selected similarity function $\mathcal{F}_n \in \mathcal{F}$. In this case, $Q'_{\text{SCS}}(\mathcal{F}_n)$ denotes the rank of the SCS value in the set$\{\text{SCS}(\mathfrak{C}(\mathcal{F}_n(\mathcal{P}_m)))|\mathcal{F}_n, \mathcal{P}_m \in \mathcal{F} \times \mathcal{P}\}$ of all SCS values and $Q'_{\text{CHI}}(\mathcal{F}_n)$ denotes the rank of the CHI value in the set $\{\text{CHI}(\mathfrak{C}(\mathcal{F}_n(\mathcal{P}_m)))|\mathcal{F}_n, \mathcal{P}_m \in \mathcal{F} \times \mathcal{P}\}$ of all CHI values. Again, for both sets applies that a larger value is assigned a larger numerical value in the ranking. The SoRC score is then defined by Equation (4.13).

$$\psi(\mathcal{F}_n(\mathcal{P}_m)) = \frac{1}{2|\mathcal{F} \times \mathcal{P}|}(Q'_{\text{SCS}}(\mathcal{F}_n) + Q'_{\text{CHI}}(\mathcal{F}_n)) \tag{4.13}$$

Theoretically, it would also be possible to calculate a single SoRC score for each combination of data set, similarity function and clustering method. However, clustering methods are often selected to serve a specific purpose. In such cases it is advisable to compute the SoRC rankings, i.e. the bar-chart-tables, separately for each clustering method of interest.

To avoid redundancies, combinations of pre-processing and analysis that do not influence each other are usually ignored and excluded from the SoRC score calculation. An example of this has been mentioned earlier, namely for the combination of scale-space representations with some similarity functions, which reduced the number of calculated clustering results from 234 to 138.

SoRC score visualization The visualization of the SoRC scores is a bar chart, which is integrated into a table structure. Thus, this visualization will be referred to as bar-chart-table. The entries of the bar-chart-tables are sorted by descending SoRC scores. Besides the SoRC score column, the visualization provides additional columns for more details. There is one column each for the normalized rank numbers of the individual cluster indices and the actual index values. A full example of the mouse urinary bladder data set $\mathcal{I}^{\mathrm{U}}$ in combination with agglomerative hierarchical clustering is presented in Figure 4.5.

4.2.6 Similarity Functions

As mentioned in the beginning, the first part of this chapter is especially interested in investigating functions to quantify the similarity between m/z-images.

So far, this chapter has discussed the importance of context for the quantification of similarity between images and assumed that this also applies to m/z-images. Furthermore, the concept of algorithmic context has been introduced in the form of pipeline setups. This concept made clear that the result of clustering depends on the interaction between the data, pre-processing, the similarity function and the clustering method. Therefore, in the context of clustering of m/z-images, the effectiveness of a similarity function to quantify the similarity between m/z-images to reveal structural relationships can be estimated by the quality of the clustering result, provided that the factors data, pre-processing, and clustering method are specified.

There are a vast number of functions that can be used to quantify the similarity between images. In this section, thirteen similarity functions are explained in more detail. The selection of functions is such that it covers a broad spectrum of different algorithmic approaches to quantify similarity.

For simplicity, each similarity function is denoted δ in the following equations. In cases where a similarity function can be computed over several image scales, the individual scale-space images $L(h, w; s\sigma^2)$ are abbreviated as $\mathcal{I}_{z;s}$. The set of all scale levels is denoted by S and its size by $|S|$. The terms in brackets are used later when SoRC is applied to real data to refer to the individual similarity functions.

Multiscale Pearson Correlation Coefficient (Pearson) The *Pearson Correlation Coefficient* is one of the most commonly used similarity functions. Since the function

is defined for one-dimensional vectors, the transformation $\mathcal{I}_z \rightarrow \mathcal{I}_z^{\mathrm{V}}$ is applied according to Equation (4.10). If a scale-space representation is computed for the m/z-images, the *Pearson Correlation Coefficient* is computed for each scale level separately and the results are averaged.

$$\begin{aligned} \delta_s(\mathcal{I}_{z;s}^{\mathrm{V}}, \mathcal{I}_{z';s}^{\mathrm{V}}) &= \frac{\sigma(\mathcal{I}_{z;s}^{\mathrm{V}}, \mathcal{I}_{z';s}^{\mathrm{V}})}{\sigma(\mathcal{I}_{z;s}^{\mathrm{V}})\sigma(\mathcal{I}_{z';s}^{\mathrm{V}})} \\ \delta &= \frac{1}{|S|} \sum_{s=0}^{|S|} \delta_s(\mathcal{I}_{z;s}^{\mathrm{V}}, \mathcal{I}_{z';s}^{\mathrm{V}}) \end{aligned} \tag{4.14}$$

where $\sigma(\cdot)$ and $\sigma(\cdot,\cdot)$ are the variance and covariance function.

Multiscale Cosine Similarity (Cosine) The *Cosine Similarity* is another commonly applied similarity function and also defined for one-dimensional vectors. So, the transformation $\mathcal{I}_z \rightarrow \mathcal{I}_z^{\mathrm{V}}$ is applied. The application in combination with scale-space representations is the same as described above for the *Pearson Correlation Coefficient*.

$$\begin{aligned} \delta_s(\mathcal{I}_{z;s}^{\mathrm{V}}, \mathcal{I}_{z';s}^{\mathrm{V}}) &= \frac{\mathcal{I}_{z;s}^{\mathrm{V}} \cdot \mathcal{I}_{z';s}^{\mathrm{V}}}{\left\|\mathcal{I}_{z;s}^{\mathrm{V}}\right\|_2 \left\|\mathcal{I}_{z';s}^{\mathrm{V}}\right\|_2} \\ \delta &= \frac{1}{|S|} \sum_{s=0}^{|S|} \delta_s(\mathcal{I}_{z;s}^{\mathrm{V}}, \mathcal{I}_{z';s}^{\mathrm{V}}) \end{aligned} \tag{4.15}$$

where $\|\cdot\|_2$ is the Euclidean norm.

Multiscale Angular Similarity (Angular) The *Angular Similarity* is similar to the *Cosine Similarity*, but more sensitive to differences between small angles. The function is also defined for one-dimensional vectors, which is why the transformation $\mathcal{I}_z \rightarrow \mathcal{I}_z^{\mathrm{V}}$ is applied. The scale-space representation handling is the same as for the *Pearson Correlation Coefficient*.

$$\begin{aligned} \delta_s(\mathcal{I}_{z;s}^{\mathrm{V}}, \mathcal{I}_{z';s}^{\mathrm{V}}) &= 1 - \left(\frac{2}{\pi} cos^{-1} \left(\frac{\mathcal{I}_{z;s}^{\mathrm{V}} \cdot \mathcal{I}_{z';s}^{\mathrm{V}}}{\left\|\mathcal{I}_{z;s}^{\mathrm{V}}\right\|_2 \left\|\mathcal{I}_{z';s}^{\mathrm{V}}\right\|_2} \right) \right) \\ \delta &= \frac{1}{|S|} \sum_{s=0}^{|S|} \delta_s(\mathcal{I}_{z;s}^{\mathrm{V}}, \mathcal{I}_{z';s}^{\mathrm{V}}) \end{aligned} \tag{4.16}$$

Equation (4.16) is only valid for $\mathcal{I}_{z;s}^{\mathrm{V}} \wedge \mathcal{I}_{z';s}^{\mathrm{V}} \in \mathbb{R}_{\geqslant 0}$.
For $\mathcal{I}_{z;s}^{\mathrm{V}} \vee \mathcal{I}_{z';s}^{\mathrm{V}} \in \mathbb{R}$ the factor $\frac{2}{\pi}$ must be changed to $\frac{1}{\pi}$.

Shared Pixel Information (Shared Pixel) The *Shared Pixel Information* was implemented as a fast pixel-to-pixel comparison function that focuses only on the actual intensity values.

$$\delta(\mathcal{I}_z, \mathcal{I}_{z'}) = 1 - \frac{\sum_{(h,w)\in\rho} |\mathcal{I}_{h,w,z} - \mathcal{I}_{h,w,z'}|}{\sum_{(h,w)\in\rho} \mathcal{I}_{h,w,z} + \sum_{(h,w)\in\rho} \mathcal{I}_{h,w,z'}} \tag{4.17}$$

Multiscale Structural Similarity Index (SSIM) The *Structural Similarity Index* is a commonly used similarity function that uses local statistics to describe and compare luminance, contrast and structure between two images and combines the result to a final similarity value. The function is defined for a pair of sliding windows, which computes a similarity map for a pair of images. The similarity map represents the local similarity between the two images at each position. Due to the sliding window approach, the local neighborhood (Moore neighborhood) of each pixel is incorporated. Arithmetic mean pooling is applied to every spectral pixel position of the similarity map to compute a single similarity value. The scale-space representation handling is the same as for the *Pearson Correlation Coefficient*.
Given an image $\mathcal{I}_{z;s}$, a pixel position $(h, w) \in p$ and a sliding window of size $o \times o$. Let $\mathfrak{W}^o_{h,w,z;s}$ denote the square matrix containing the values of the $o \times o$ sized sliding window centered at (h, w).

$$\begin{aligned}
&\kappa(\mathcal{I}_{z;s}, \mathcal{I}_{z';s})_{h,w} \\
&\quad = \frac{(2\mu(\mathfrak{W}^o_{h,w,z;s})\mu(\mathfrak{W}^o_{h,w,z';s}) + c_1)(2\sigma(\mathfrak{W}^o_{h,w,z;s}, \mathfrak{W}^o_{h,w,z';s}) + c_2)}{(\mu^2(\mathfrak{W}^o_{h,w,z;s}) + \mu^2(\mathfrak{W}^o_{h,w,z';s}) + c_1)(\sigma^2(\mathfrak{W}^o_{h,w,z;s}) + \sigma^2(\mathfrak{W}^o_{h,w,z';s}) + c_2)} \\
&\delta_s(\mathcal{I}_{z;s}, \mathcal{I}_{z';s}) = \frac{1}{|\rho|} \sum_{(h,w)\in\rho} \kappa(\mathcal{I}_{z;s}, \mathcal{I}_{z';s})_{h,w} \\
&\delta = \frac{1}{|S|} \sum_{s=0}^{|S|} \delta_s(\mathcal{I}_{z;s}, \mathcal{I}_{z';s})
\end{aligned} \tag{4.18}$$

where κ is a function that takes two sliding windows and moves them across two images. The window size was set to $o = 13$. The function returns a new image, which is referred to as a similarity map. $\mu(\cdot)$ and $\sigma(\cdot)$ are functions that calculate the arithmetic mean and the standard deviation of the sliding window, c_1, c_2 are two small constants to avoid division by zero. $|\rho|$ is the size of the set of all spectral pixel positions (see Section 1.1.1).

Multifeature Similarity Index (MFS) Following the concept of the SSIM, we created the *Multifeature Similarity Index*. The MFS follows the basic formula of the SSIM and is defined between two sliding windows. However, it does not only consider the pixel intensity values but also the luminance, contrast and structure features of the orientation images and the magnitude images (see Equations (4.8) and (4.9)). For each pair of m/z-images, three similarity maps are computed and combined. For the combination into a single similarity map, a pooling function is applied at each pixel position. In order to consider only the most dominant features, maximization is selected as the pooling function. Another interesting pooling option, which considers every feature, is the arithmetic mean. However, previous experiments showed worse results than maximization in almost all cases. Again, the final similarity map has to be pooled to compute a single similarity value. Like in the SSIM, the arithmetic mean is selected. The scale-space representation handling is the same as for the *Pearson Correlation Coefficient*.
Similar to the SSIM, $\mathfrak{W}^o_{h,w,z;s}$ denotes a square matrix containing the values of an $o \times o$ sized sliding window centered at (h, w).

$$
\begin{aligned}
\kappa_\mu(\mathcal{I}_{z;s}, \mathcal{I}_{z';s})_{h,w} &= \frac{2\mu(\mathfrak{W}^o_{h,w,z;s})\mu(\mathfrak{W}^o_{h,w,z';s}) + c}{\mu(\mathfrak{W}^o_{h,w,z;s})^2 + \mu(\mathfrak{W}^o_{h,w,z';s})^2 + c} \\
\kappa_\sigma(\mathcal{I}_{z;s}, \mathcal{I}_{z';s})_{h,w} &= \frac{2\sigma(\mathfrak{W}^o_{h,w,z;s})\sigma(\mathfrak{W}^o_{h,w,z';s}) + c}{\sigma(\mathfrak{W}^o_{h,w,z;s})^2 + \sigma(\mathfrak{W}^o_{h,w,z';s})^2 + c} \\
\kappa_{cov}(\mathcal{I}_{z;s}, \mathcal{I}_{z';s})_{h,w} &= \frac{\sigma(\mathfrak{W}^o_{h,w,z;s}, \mathfrak{W}^o_{h,w,z';s}) + c}{\sigma(\mathfrak{W}^o_{h,w,z;s})\sigma(\mathfrak{W}^o_{h,w,z';s}) + c} \\
\Xi_I &= \kappa_\mu(\mathcal{I}_{z;s}, \mathcal{I}_{z';s})_{h,w}\ \kappa_\sigma(\mathcal{I}_{z;s}, \mathcal{I}_{z';s})_{h,w}\ \kappa_{cov}(\mathcal{I}_{z;s}, \mathcal{I}_{z';s})_{h,w} \\
\Xi_O &= \kappa_\mu(\mathcal{I}^{\mathrm{G}}_{z;s}, \mathcal{I}^{\mathrm{G}}_{z';s})_{h,w}\ \kappa_\sigma(\mathcal{I}^{\mathrm{G}}_{z;s}, \mathcal{I}^{\mathrm{G}}_{z';s})_{h,w}\ \kappa_{cov}(\mathcal{I}^{\mathrm{G}}_{z;s}, \mathcal{I}^{\mathrm{G}}_{z';s})_{h,w} \\
\Xi_M &= \kappa_\mu(\mathcal{I}^{\mathrm{M}}_{z;s}, \mathcal{I}^{\mathrm{M}}_{z';s})_{h,w}\ \kappa_\sigma(\mathcal{I}^{\mathrm{M}}_{z;s}, \mathcal{I}^{\mathrm{M}}_{z';s})_{h,w}\ \kappa_{cov}(\mathcal{I}^{\mathrm{M}}_{z;s}, \mathcal{I}^{\mathrm{M}}_{z';s})_{h,w} \\
\Xi_{I'} &= \mathrm{sign}(\Xi_I)|\Xi_I|^{\frac{1}{3}} \\
\Xi_{O'} &= \mathrm{sign}(\Xi_O)|\Xi_O|^{\frac{1}{3}} \\
\Xi_{M'} &= \mathrm{sign}(\Xi_M)|\Xi_M|^{\frac{1}{3}} \\
\Xi_W &= \max_{(h,w)\in\rho}(\Xi_{I_{h,w}}, \Xi_{O_{h,w}}, \Xi_{M_{h,w}}) \\
\delta_s(\mathcal{I}_{z;s}, \mathcal{I}_{z';s}) &= \frac{1}{|\rho|}\sum_{(h,w)\in\rho}\Xi_{W_{h,w}}; \\
\delta &= \frac{1}{|S|}\sum_{s=0}^{|S|}\delta_s(\mathcal{I}_{z;s}, \mathcal{I}_{z';s})
\end{aligned}
\tag{4.19}
$$

where κ_μ, κ_σ and κ_{cov} are functions that take two sliding windows and moves them across two images. The window size was set to $o = 13$. Like κ in Equation (4.18) these functions return similarity maps. $\mu(\cdot)$, $\sigma(\cdot)$ and $\sigma(\cdot,\cdot)$ are functions that calculate the arithmetic mean, standard deviation and covariance of the sliding window. sign$(\cdot)$ returns the signs of its input, $|\Xi|$ represents the absolute values of Ξ and c is a small constant to avoid division by zero.

Local Standard Deviation based Image Quality Index (Local Std) An adjusted version of the *Local Standard Deviation based Image Quality Index* [35] is included to investigate the effect of an intensity deviation based approach.

$$\begin{aligned} \delta(\mathcal{I}_z, \mathcal{I}_{z'}) &= \frac{1}{|\rho|} \sum_{(h,w)\in\rho} \mathcal{I}_\sigma \\ \mathcal{I}_\sigma &= \frac{2\theta(\mathcal{I}_z)\theta(\mathcal{I}'_z) + c}{\theta(\mathcal{I}_z)^2 + \theta(\mathcal{I}'_z)^2 + c} \end{aligned} \tag{4.20}$$

where $\theta(\mathcal{I}_z)$ is the result of the convolution of $\mathcal{I}_z$ with a standard deviation kernel of size 13 and c is a small constant to avoid division by zero.

Intensity-Magnitude-Angle Similarity (IMA Sim) The *Intensity-Magnitude-Angle Similarity* is based on a similar idea as the *Multifeature Similarity Index*, but it is based on a pixel-to-pixel comparison instead of a sliding window approach. Therefore, the function is faster to compute, but no local neighborhood information is incorporated (except for the gradient and magnitude information).

$$\begin{aligned} \delta(\mathcal{I}_z, \mathcal{I}_{z'}) &= \frac{1}{|\rho|} \sum_{(h,w)\in\rho} \frac{1}{3}\delta_{\mathrm{i}}(\mathcal{I}_z, \mathcal{I}_{z'}) + \frac{1}{3}\delta_{\mathrm{m}}(\mathcal{I}_{z'}, \mathcal{I}_{z'}) + \frac{1}{3}\frac{\delta_{\mathrm{a}}(\mathcal{I}_z, \mathcal{I}_{z'}) + 1}{2} \\ \delta_{\mathrm{i}}(\mathcal{I}_z, \mathcal{I}_{z'}) &= \frac{2\mathcal{I}_z\mathcal{I}_{z'} + c}{{\mathcal{I}_z}^2 + {\mathcal{I}_{z'}}^2 + c} \\ \delta_{\mathrm{m}}(\mathcal{I}_z, \mathcal{I}_{z'}) &= \frac{2\,\mathcal{I}^{\mathrm{M}}_z\mathcal{I}^{\mathrm{M}}_{z'}| + c}{(\mathcal{I}^{\mathrm{M}}_z)^2 + (\mathcal{I}^{\mathrm{M}}_{z'})^2 + c} \\ \delta_{\mathrm{a}}(\mathcal{I}_z, \mathcal{I}_{z'}) &= \mathcal{I}^{\nabla}_z \cdot \mathcal{I}^{\nabla}_{z'} \end{aligned} \tag{4.21}$$

Contingency Similarity Index (Contingency) The *Contingency Similarity Index* was implemented to investigate the applicability of multi-thresholding to quantify the similarity between m/z-images.

$$
\begin{aligned}
c_{\mathrm{l}}(\mathcal{I}_z) &= 0.2 \max(\mathcal{I}_z) \\
c_{\mathrm{u}}(\mathcal{I}_z) &= 0.8 \max(\mathcal{I}_z) \\
\mathcal{I}^c_{h,w,z} &= \begin{cases} 1, & \text{if } \mathcal{I}_{h,w,z} < c_{\mathrm{l}}(\mathcal{I}_z) \wedge (h,w) \in \rho \\ 2, & c_{\mathrm{l}}(\mathcal{I}_z) \leqslant \mathcal{I}_{h,w,z} \leqslant c_{\mathrm{u}}(\mathcal{I}_z) \wedge (h,w) \in \rho \\ 3, & \text{if } \mathcal{I}_{h,w,z} > c_{\mathrm{u}}(\mathcal{I}_z) \wedge (h,w) \in \rho \\ 0, & \text{if } (h,w) \notin \rho \end{cases} \\
\mathcal{U} &= \mathfrak{U}_{\{1,2,3\}}(\mathcal{I}^c_z, \mathcal{I}^c_{z'}) \\
\delta(\mathcal{I}_z, \mathcal{I}_{z'}) &= \frac{1}{\sum \mathcal{U}} \big((\mathcal{U}_{0,0} + \mathcal{U}_{1,1} + \mathcal{U}_{2,2}) \\
&\quad - (\mathcal{U}_{0,1} + \mathcal{U}_{1,2} + \mathcal{U}_{1,0} + \mathcal{U}_{2,1}) \\
&\quad - (\mathcal{U}_{0,2} + \mathcal{U}_{2,0}) \big)
\end{aligned}
\tag{4.22}
$$

where $\mathfrak{U}_{\{1,2,3\}}(\cdot,\cdot)$ calculates a 3×3 contingency table of two images according to the classes $\{1,2,3\}$.

Gradient Information (Grad Info) The *Gradient Information* function is included as a function that considers only the gradient information, i.e. the gradient direction and gradient magnitude. So this function makes no direct use of pixel intensity values.

$$
\begin{aligned}
\delta(\mathcal{I}_z, \mathcal{I}_{z'}) &= \frac{1}{|\rho|} \sum_{(h,w) \in \rho} \delta' \min(\mathcal{I}^{\mathrm{M}}_z, \mathcal{I}^{\mathrm{M}}_{z'}) \\
\delta' &= \frac{1}{2} \left(\frac{\mathcal{I}^{\nabla}_z \cdot \mathcal{I}^{\nabla}_{z'}}{\mathcal{I}^{\mathrm{M}}_z \mathcal{I}^{\mathrm{M}}_{z'}} + 1 \right)
\end{aligned}
\tag{4.23}
$$

Hypergeometric Similarity Measure (Hypergeometric) Since a previous study reported very good results of the *Hypergeometric Similarity Measure* [48], we tested this measure as well. However, we implemented a slightly modified version that achieved better results on our data compared to the originally proposed formula.

$$
\begin{aligned}
\mathfrak{B}_z &= \sum \text{binarize}(\mathcal{I}_z) \\
\mathfrak{B}_{z'} &= \sum \text{binarize}(\mathcal{I}_{z'}) \\
\mathfrak{B}_{z,z'} &= \sum (\text{binarize}(\mathcal{I}_z) \wedge \text{binarize}(\mathcal{I}_{z'})) \\
\beta_1 &= \frac{|\rho| - \mathfrak{B}_{z'}}{|\rho|} \\
\alpha_1 &= \frac{\mathfrak{B}_z - \mathfrak{B}_{z,z'}}{\mathfrak{B}_z} - \beta_1 \\
\beta_2 &= \frac{\mathfrak{B}_{z'}}{|\rho|} \\
\alpha_2 &= \frac{\mathfrak{B}_{z,z'}}{\mathfrak{B}_z} - \beta_2 \\
\hat{\alpha} &= \begin{cases} \frac{\beta_1}{\beta_1+\alpha_1}^{\beta_1+\alpha_1} \frac{1-\beta_1}{1-\beta_1-\alpha_1}^{1-\beta_1-\alpha_1{}^2}, & \text{if } 1-\beta_1-\alpha_1 > 0 \wedge \beta_1+\alpha_1 > 0 \\ \infty, & \text{otherwise} \end{cases} \\
\hat{\beta} &= \begin{cases} \frac{\beta_2}{\beta_2+\alpha_2}^{\beta_2+\alpha_2} \frac{1-\beta_2}{1-\beta_2-\alpha_2}^{1-\beta_2-\alpha_2{}^2}, & \text{if } 1-\beta_2-\alpha_2 > 0 \wedge \beta_2+\alpha_2 > 0 \\ \infty, & \text{otherwise} \end{cases} \\
\delta' &= \begin{cases} 1, & \text{if } \hat{\alpha} - \hat{\beta} = -\infty \\ -1, & \text{if } \hat{\alpha} - \hat{\beta} = \infty \\ \hat{\alpha} - \hat{\beta}, & \text{otherwise} \end{cases} \\
\delta(\mathcal{I}_z, \mathcal{I}_{z'}) &= \frac{1}{2}(\delta'(\mathcal{I}_z, \mathcal{I}_{z'}) + \delta'(\mathcal{I}_{z'}, \mathcal{I}_z))
\end{aligned}
\tag{4.24}
$$

where binarize$(\cdot)$ is a binarization function.

Intensity Histogram Similarity (Histogram) The *Intensity Histogram Similarity* compares intensity value histograms ($\mathcal{I}_z^{\mathrm{P}}$) of m/z-images (see Section 4.2.2) using the Hellinger distance [41]. Accordingly, the function uses only so-called global features and discards any spatial information between the pixels, even the relative positioning. The Hellinger distance is used for histogram comparison as it is directly related to the Euclidean norm, which supports an intuitive understanding of the comparison

approach. However, there are numerous other functions for comparing histograms which could also be used.

$$\begin{aligned} \delta(\mathcal{I}_z, \mathcal{I}_{z'}) &= 1 - \mathrm{HD}(\mathcal{I}_z, \mathcal{I}_{z'}) \\ HD(\mathcal{I}_z, \mathcal{I}_{z'}) &= \frac{1}{\sqrt{2}} \sqrt{\sum_{\iota} \left(\sqrt{\mathcal{I}^{\mathrm{P}}_{z,\iota}} - \sqrt{\mathcal{I}^{\mathrm{P}}_{z',\iota}} \right)^2} \end{aligned} \tag{4.25}$$

where $\mathrm{HD}(\cdot)$ is the Hellinger distance, ι describes the binning of $\mathcal{I}^{\mathrm{P}}_z$ and $\mathcal{I}^{\mathrm{P}}_z$ is the representation of an m/z-image as one-dimensional intensity histogram according to Section 4.2.2.

Mutual Information (Mutual Info) The *Mutual Information* is a commonly known information theoretical measure that quantifies the "amount of information". It is included since information theoretical approaches are regularly used in many different areas of signal analysis [92].

$$\begin{aligned} \delta(\mathcal{I}_z, \mathcal{I}_{z'}) &= H(\mathcal{I}_z) + H(\mathcal{I}_{z'}) - H(\mathcal{I}_z, \mathcal{I}_{z'}) \\ H(\mathcal{I}_z) &= -\sum_{\iota} \mathcal{I}^{\mathrm{P}}_{z,\iota} log_e \mathcal{I}^{\mathrm{P}}_{z,\iota} \\ H(\mathcal{I}_z, \mathcal{I}_{z'}) &= -\sum_{\iota} \mathcal{I}^{\mathrm{P}}_{(z,z'),\iota} log_e \mathcal{I}^{\mathrm{P}}_{(z,z'),\iota} \end{aligned} \tag{4.26}$$

where ι describes the binning of $\mathcal{I}^{\mathrm{P}}_z$ and $\mathcal{I}^{\mathrm{P}}_{(z,z')}$, and $\mathcal{I}^{\mathrm{P}}_{(z,z')}$ is the representation of two m/z-images as a one-dimensional joint intensity histogram.

Summary The selection of similarity functions presented above includes commonly known and frequently used functions in the field of MSI (*Pearson Correlation Coefficient*, *Cosine Similarity*, *Angular Similarity*), functions that consider local neighborhoods within the spatial domain (*Structural Similarity Index*), combined with alternative image features (*Multifeature Similarity Index*), direct pixel-to-pixel comparisons, with and without consideration of alternative image features (*Shared Pixel Information*, *Intensity-Magnitude-Angle Similarity*), functions that consider multi-thresholds (*Contingency Similarity Index*) and local variations (*Local Standard Deviation based Image Quality Index*), and functions that omit actual pixel intensity values and use gradients instead (*Gradient Information*). Furthermore, functions based on the comparison of distributions (*Hypergeometric Similarity Index*) or global features (*Intensity Histogram Similarity*) and an information theoretical approach (*Mutual Information*) are included.
The implementation of SoRC contains many more similarity functions. However, the presented selection is composed in a way that both the most competitive functions

(based on previous experiments) have been included and a wide range of different approaches is covered.

4.2.7 Clustering Methods

To consider not only the influence of the similarity function on the clustering result but also the influence of the clustering method, three different methods are used in the application on real data below.

1. Agglomerative Hierarchical Clustering [112]
2. Affinity Propagation [34]
3. Community Detection on Mass Channel Similarity Graphs

All three methods are explained briefly in the following.

Hierarchical Clustering Hierarchical clustering is very prominent in bioinformatics and is used in a variety of different fields. A major advantage of this method is that it allows to track each clustering step by inspecting the tree-like result called a dendrogram. The term hierarchical clustering itself is an umbrella term for a whole family of methods. In this section, the agglomerative average linkage (UPGMA) method is applied [112]. The idea behind agglomerative average linkage hierarchical clustering is to initialize each data point (i.e. m/z-image) as a single cluster. Thereafter, the two clusters with the closest average distance according to a similarity criterion are merged. This procedure is applied iteratively until only a single cluster remains or until the intended number of clusters is reached. Without the use of approximation techniques, hierarchical clustering requires that the number of clusters is determined manually. To avoid an arbitrary choice, a universally applicable approximation technique is used.

$$\varsigma = \mu(\Upsilon) + (c\,\sigma(\Upsilon)) \tag{4.27}$$

Υ represents the set of all distances between two clusters that were merged during clustering until only one cluster is left. μ and σ are the arithmetic mean and the standard deviation and c is a tuning factor, which was set to one. The number of clusters is determined as the smallest possible number, where no data point has a cophenetic distance greater than ς. The cophenetic distance between two data points is defined as the distance between the two clusters that contain these data points at the location in the dendrogram where they are merged.

Affinity Propagation The idea behind affinity propagation is to determine a set of data points that are suited to represent the entire data. To identify those points, messages are sent between pairs of data points until convergence. Each message belongs to one of two categories. First, the "responsibility", which describes how well a data item m is suited as a representative for another data point n. Second, the "availability", which describes how appropriate it would be for a data point n to pick another data point m as its representative [34]. The method detects the number of clusters automatically. The applied implementation requires three parameters: the convergence iteration (set to 15), the maximum number of iterations (set to 200) and the damping factor (set to 0.5).

Community Detection on Mass Channel Similarity Graphs The basic idea of this method is to create a similarity graph from the set of all m/z-images, the so-called mass channel similarity graph, and to use community detection as a clustering method. The basic outline of this approach is shown in Figure 4.4. Further details follow in Section 4.3 and Section 4.3.1 using edge reduction method ER-STAT.

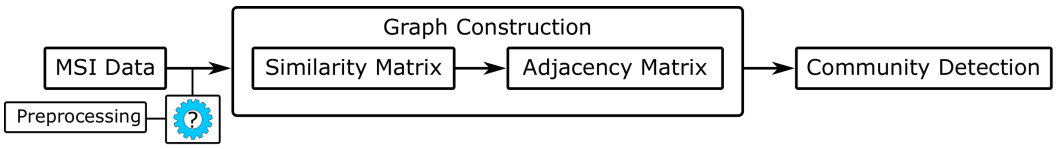


Fig. 4.4.: Outline for the community detection approach on m/z-images – To use community detection, a graph representation of the relationship between m/z-images is required. For this purpose, a pairwise similarity matrix of all m/z-images is computed. A thresholding method is applied to convert the similarity matrix into an adjacency matrix, which in turn determines the graph structure. The applied thresholding procedure is similar to Equation (4.27). However, in this case, Υ equals the set of all similarity values of the similarity matrix. To compute the clusters, a community detection method is applied to the graph. The number of clusters is determined by the applied method.

All three clustering methods detect the number of clusters automatically. Therefore, the number of clusters can differ for each pipeline setup. However, since the number of clusters is directly related to the structural properties of the data and the pipeline setup, it can be considered as a characteristic feature of the result.

Besides the three clustering methods above, the SoRC implementation supports k-medoids (partitioning around medoids (PAM) version [83] and expectation-maximization (EM) version [44]), DBSCAN [29], OPTICS [8] and Spectral clustering [107]. However, the implementation can easily be extended to include any type of clustering method, as long as the labels are adjusted to start with index one and noise labels, if they exist, are assigned to unique numbers.

4.2.8 Application on Real Data

The following presents an example of an application of the SoRC workflow to real data. The SoRC workflow is applied as illustrated in Figure 4.3. The processed variants of all data sets presented in Chapter 2 are used, i.e. $\mathcal{I}^{\mathrm{B}}$, $\mathcal{I}^{\mathrm{U}}$, $\mathcal{I}^{\mathrm{K}}$ and $\mathcal{I}^{\mathrm{S}}$.
It is commonly known that the MSI technology suffers from a large variation of different intensity value ranges between individual m/z-images. To compensate for this, an initial min-max normalization was used to scale each m/z-image linearly to a range of $[0, 1]$. Then, a ProViM module was used to write the individual m/z-images to disk using the "viridis" colormap. For the evaluation with the SoRC workflow, these images were read in again. Due to the read and write operations with the "viridis" colormap the range shifted into $[0.11765, 0.84314]$. Due to the min-max normalization, the following results are only valid in this context and should not be transferred directly to the original data sets.

Specification of the applied pre-processing To investigate the influence of pre-processing on the quantification of the similarity between m/z-images and the cluster formation, a very general procedure is applied. This pre-processing procedure was referred to as P_1 in Section 4.2.4.

1. Zero-padded with a padding width of thirteen pixels.
2. Removal of low-intensity signals, using Otsu's method for threshold determination [85].
3. Gaussian smoothing with a Gaussian kernel of size $\sigma = 0.8$.

Zero-padding allows a consistent and controlled behavior for each similarity function at the m/z-image borders. Therefore, it is also applied for the pipeline setups no-preprocessing, $\mathrm{P}_{2\mathrm{a}}$ and $\mathrm{P}_{2\mathrm{b}}$. The removal of signals below a certain threshold reduces the amount of noise signals. Otsu's method is a commonly used and reliable method to detect such a threshold. The additional Gaussian smoothing is applied to reduce the strong pixel-to-pixel signal variation inherent to some MSI technologies, such as MALDI-MSI.

Remark This thesis does not have the ambition to find the best pre-processing method to enhance m/z-images. Besides, this would depend strongly on the context types mentioned above anyway, as well as on the research question and the objective of the analysis. In order to optimize the pre-processing, the analysis of different pipeline setups would also be useful. However, this would go beyond the scope of

this thesis. Instead, the applied pre-processing has been selected to address some of the most common problems associated with m/z-images.

Specification of the applied scale-space representation To investigate the potential benefits of scale-space representations, three different scale-space representations are computed according to Equation (4.11) in Section 4.2.3 using $s \in \{0\}$, $s \in \{0, 1, 2\}$ (P_{2a}) and $s \in \{0, 2, 4\}$ (P_{2b}). The idea behind using different scale levels is to investigate the assumption made in Section 4.2.3, i.e. the original scale should be more suited to compare finely detailed structures like edges, corners and hot spots, while larger scales should be more suited to compare coarser structures like homogeneous regions. The omission of a scale-space representation, i.e. $s = \{0\}$, is used as a control to investigate whether the use of scale-space representations offers any benefit at all for the analysis of m/z-images. The other two scale-space representations, i.e. P_{2a} and P_{2b}, are used to investigate whether there is a potential benefit in using different scale-space levels.

Results

As mentioned before, the implementation of the SoRC workflow as used in this thesis evaluates the pipeline setups separately for each combination of data set and clustering method. Consequently, the workflow generates a separate bar-chart-table visualization for each combination of data set and clustering method. This means, that a single bar-chart-table compares the pipeline setups for combinations of the six pre-processing variants with the thirteen similarity functions.

Before all results are compared with each other, an example of the entire evaluation procedure is presented using the agglomerative hierarchical clustering results for the mouse urinary bladder data set ($\mathcal{I}^{U}$). Figure 4.5 shows the bar-chart-table visualization of the 30 highest SoRC scores (out of a total of 46) for the different combinations in pre-processing and similarity function, but with a fixed clustering method. It can be seen that in these 30 examples, functions with pre-processing always score better than their counterparts without pre-processing. For those similarity functions where the integration of a scale-space representation is available, the larger step size scores better than the smaller step size, which scores better than no utilization of the scale-space representation. The evaluation of the individual columns for the SCS and the CHI shows that the indices "disagree" for some combinations, i.e. one index rates the clustering result better than the other. This demonstrates the importance of using at least two not strictly correlated cluster indices that assess different characteristics

of a clustering. In general, the SoRC score is higher if both indices rate the clustering similarly. The actual index values, presented in the last two columns, can be used to either examine the quality difference between two pipeline setups in more detail or to compare the bar-chart-tables for different clustering methods.

To include the different clustering methods in the analysis, multiple bar-chart-tables can be compared, which is shown in Figure 4.6. Figure 4.6 shows a comparison of the SoRC scores for each combination of data set and clustering method considered in this section. For clarity, the columns of the bar-chart-table are reduced to the SoRC scores and the rows are reduced to the five highest-scoring pipeline setup combinations.

The evaluation of the comparison yields six major observations:

1. The *Cosine Similarity* and/or the *Pearson Correlation Coefficient* are among the best scoring similarity functions for most combinations of data set and clustering method. The only exception is the human skin sample ($\mathcal{I}^{\mathrm{S}}$).

2. For $\mathcal{I}^{\mathrm{B}}$, $\mathcal{I}^{\mathrm{U}}$ and $\mathcal{I}^{\mathrm{K}}$, most similarity functions achieve better scores in combination with pre-processing. There is only one exception for the Histogram function for $\mathcal{I}^{\mathrm{K}}$, combined with community detection as the clustering method. However, in $\mathcal{I}^{\mathrm{S}}$ the scores are more mixed.

3. The comparison indicates that most pipeline setups achieve better scores when they include one of the scale-space representations, which is only available for Pearson, Cosine, Angular, SSIM and MFS. Only one combination benefits from the omission of the scale-space representation, which is the Pearson function for $\mathcal{I}^{\mathrm{K}}$, combined with affinity propagation as the clustering method. Additionally, there are a few examples where at least one of the step sizes performs worse than the omission. Furthermore, there is no clear result whether the smaller or the larger step size performs better.

4. The five highest SoRC scores for $\mathcal{I}^{\mathrm{B}}$, $\mathcal{I}^{\mathrm{U}}$ and $\mathcal{I}^{\mathrm{K}}$ are higher than for $\mathcal{I}^{\mathrm{S}}$ in almost all cases, with one exception that is the Contingency function for $\mathcal{I}^{\mathrm{S}}$ in combination with agglomerative hierarchical clustering.

5. Global image features, which discard any local features achieve inferior results.

6. The last and probably most important observation is that no pipeline setup always leads to the best score.

Table 4.1 shows a summary of the time requirements needed to evaluate each data set with the SoRC workflow. Each of the six pre-processing setups was executed in

Method	SCS Rank	CHI Rank	Score	SCS Val	CHI Val
PP - 2 MS - Cosine	0.978261	0.902174	0.940217	0.522151	20.427
PP - 1 MS - Cosine	0.934783	0.902174	0.918478	0.512069	20.427
PP - 2 MS - Pearson	1	0.76087	0.880435	0.536637	18.507
PP - 0 MS - Shared Pixel	0.73913	0.978261	0.858696	0.355478	23.2021
PP - 1 MS - Pearson	0.956522	0.673913	0.815217	0.515357	16.6129
PP - 0 MS - Pearson	0.913043	0.695652	0.804348	0.489046	16.6402
PP - 0 MS - Contingency	0.586957	1	0.793478	0.324005	24.6681
PP - 2 MS - Angular	0.717391	0.869565	0.793478	0.352995	20.4123
PP - 0 MS - Cosine	0.847826	0.73913	0.793478	0.442377	18.2812
PP - 1 MS - Angular	0.673913	0.847826	0.76087	0.34266	20.2471
PP - 2 MS - SSIM	0.456522	0.956522	0.706522	0.306237	21.6991
PP - 0 MS - Local Std	0.76087	0.630435	0.695652	0.367423	13.9116
PP - 0 MS - Hypergeometric	0.630435	0.717391	0.673913	0.339743	16.7326
2 MS - Cosine	0.804348	0.48913	0.646739	0.390586	10.4542
PP - 1 MS - SSIM	0.347826	0.934783	0.641304	0.296903	20.5879
1 MS - Cosine	0.782609	0.48913	0.63587	0.385208	10.4542
PP - 0 MS - Angular	0.478261	0.782609	0.630435	0.308765	19.4556
PP - 0 MS - Grad Info	0.695652	0.565217	0.630435	0.351819	12.4117
2 MS - Pearson	0.891304	0.347826	0.619565	0.458023	9.56043
1 MS - Pearson	0.869565	0.369565	0.619565	0.443494	9.56043
PP - 0 MS - IMA Sim	0.304348	0.804348	0.554348	0.283237	20.0057
0 MS - Local Std	0.826087	0.195652	0.51087	0.399586	7.69302
0 MS - Shared Pixel	0.5	0.521739	0.51087	0.312273	11.2394
PP - 2 MS - MFS	0.434783	0.586957	0.51087	0.3049	12.741
PP - 0 MS - SSIM	0.152174	0.826087	0.48913	0.247174	20.227
PP - 1 MS - MFS	0.369565	0.608696	0.48913	0.299497	13.0265
PP - 0 MS - MFS	0.413043	0.543478	0.478261	0.301898	12.1403
0 MS - Pearson	0.652174	0.304348	0.478261	0.341598	9.48525
PP - 0 MS - Histogram	0.195652	0.652174	0.423913	0.262931	14.2257
0 MS - Hypergeometric	0.543478	0.26087	0.402174	0.319135	8.53329

Fig. 4.5.: SoRC score bar-chart-table visualization for the mouse urinary bladder data set $\mathcal{I}^{\mathrm{U}}$ combined with agglomerative hierarchical clustering – The bar-chart-table visualization shows the 30 highest SoRC scores (out of 46). The **Method** column contains the applied pipeline setups. PP indicates that pre-processing was applied and 0MS, 1MS and 2MS indicate the use of scale-space representations with $s = \{0\}$, $s = \{0, 1, 2\}$ ($\mathrm{P}_{2\mathrm{a}}$) and $s = \{0, 2, 4\}$ ($\mathrm{P}_{2\mathrm{b}}$), respectively. The columns **SCS Rank** and **CHI Rank** contain the normalized ranks, while the columns **SCS Val** and **CHI Val** contain the actual index values. The whole bar-chart-table is sorted according to the SoRC scores, which are shown in the **Score** column.

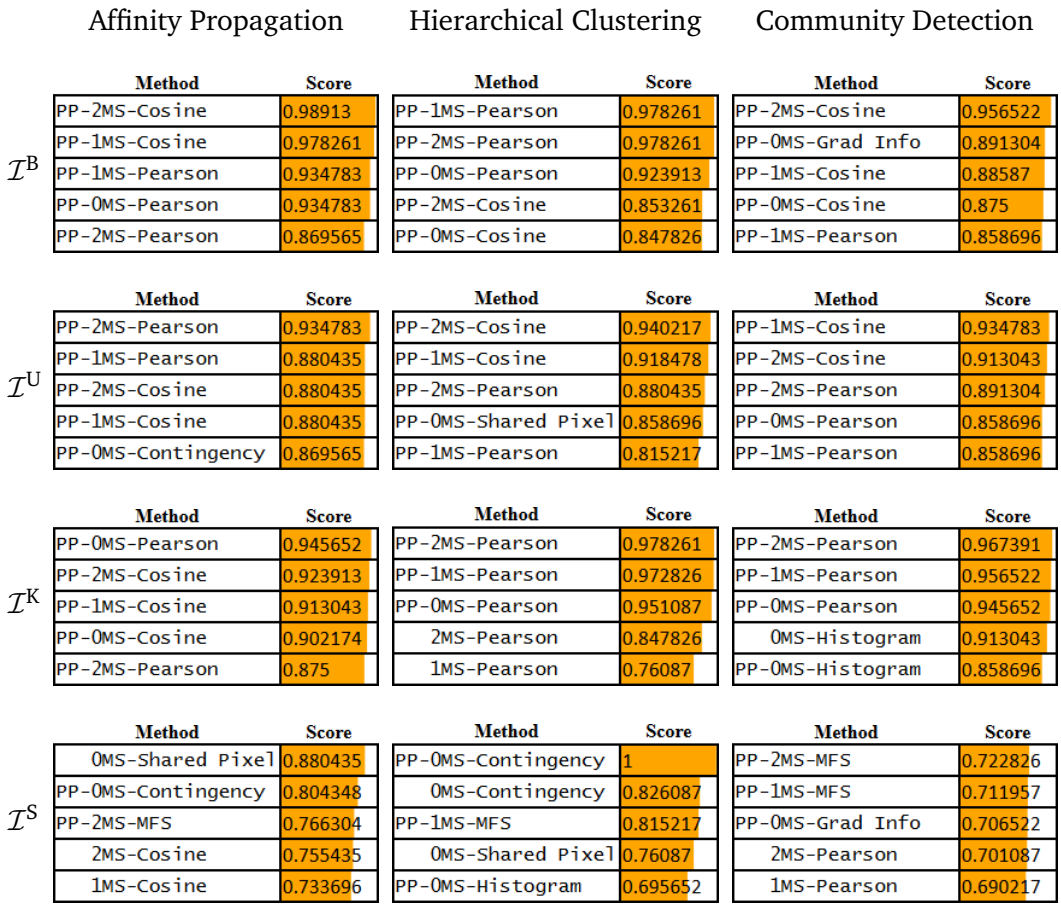


Fig. 4.6.: Comparison of the SoRC scores for each combination of data set and clustering method considered in Section 4.2.8 – The table-bar-charts are reduced to the pipeline setup combinations with the five highest SoRC scores. `PP` indicates that pre-processing was applied and `0MS`, `1MS` and `2MS` indicate the use of scale-space representations with $s = \{0\}$, $s = \{0, 1, 2\}$ (P_{2a}) and $s = \{0, 2, 4\}$ (P_{2b}), respectively. A higher score at the same position between the table-bar-charts does not necessarily imply a better result. It rather indicates that there are fewer numerical ties and fewer "disagreements" between SCS and CHI.

Tab. 4.1.: Time requirements of the SoRC workflow – Each pre-processing setup was executed in parallel on a machine with 500 GB memory and 28 CPUs. **Total** illustrates the time requirement if the individual setups are not executed in parallel.

Pre-processing setup	$\mathcal{I}^{\mathrm{B}}$	$\mathcal{I}^{\mathrm{U}}$	$\mathcal{I}^{\mathrm{K}}$	$\mathcal{I}^{\mathrm{S}}$
0 MS	1:08:58	4:05:14	5:22:21	1:00:43
1 MS	1:20:42	4:36:36	6:39:38	1:08:46
2 MS	1:21:21	4:36:52	6:39:45	1:10:47
PP – 0 MS	1:08:04	3:17:23	4:42:38	0:42:36
PP – 1 MS	1:19:03	3:57:17	5:31:17	0:50:53
PP – 2 MS	1:19:59	3:57:14	6:05:06	0:50:47
Total	7:38:07	24:30:36	35:00:45	5:44:32

parallel. It can be seen that with an increasing number and size of the images the time requirements increase notably. However, this table is only intended to give an idea of the potential time required, as the time required for each individual pipeline setup can vary significantly, e.g. the *Pearson Correlation Coefficient* is computed much faster than the *Multifeature Similarity Index*.

Since the specific implementation of the SoRC workflow is flexible and the results cannot be generalized across data sets, the presented results are only valid for the workflow implementation as described above and the given data sets. However, due to the classification of the data sets in the regularity scale, assumptions about potential generalizations can be made.

The observations resulting from the comparison indicate that the use of pre-processing and scale-space representations seem to have a positive impact on the final clustering results for the data sets used in this thesis, independent from the clustering method or the similarity function.
This observation can be explained by the well-known problems that are associated with m/z-images, such as pixel-to-pixel variation and the occurrence of intensity hotspots. The influence of most of these problems can be reduced by pre-processing, as applied in P_1, or by similar procedures. However, the results also indicate that pre-processing can have a negative influence on molecular distributions that are either small or finely structured. There is a chance that finely structured patterns will become too blurred or disappear completely. The applied pre-processing procedure is not very intense, which might explain why it does not negatively affect the result of $\mathcal{I}^{\mathrm{U}}$, which has some finely structured morphologies. Another reason may be the fact that the m/z-images showing these finely structured morphologies represent only a small subset of the complete data set. In contrast, $\mathcal{I}^{\mathrm{S}}$ has significantly more

m/z-images, which either have rather small or finely structured intensity distributions. In summary, pre-processing seems to be especially beneficial for samples with high regularity, but it should be used with caution for samples with low regularity as it could have negative effects. Considering the amount of different influences that different pre-processing procedures can have on the m/z-images, it can be concluded that there will be no perfect procedures and that pre-processing should be adjusted to the data set and the analysis objective.
A similar conclusion applies to the use of scale-space representations. This is not unexpected, since the scale-space representations are computed by repeated smoothing, which is also a kind of pre-processing. The idea to combine multiple scale levels to consider fine-grained structures at lower scales (less smoothing) and coarser structures at higher scales (more smoothing) seems to work out well in most cases. The results presented suggest that the use of scale-space representations should be considered more often in the analysis of m/z-images. An integration of scale-space representations as presented in this section, i.e. by combining the results of several scale levels with an aggregation function, can be used in combination with any similarity function and should also be applicable to a variety of other analyses. However, the results also show that combinations of different scale levels should be considered, since not every combination of scale levels works equally well for every data set. Furthermore, based on the results presented here, no generalization for the combination of scale levels was possible.

In the literature, we have not found a comprehensive comparative analysis of similarity functions for application to m/z-images considering different degrees of regularity. Therefore, a strong focus of this section was the comparison of different similarity functions. The presented results indicate that two of the most commonly used similarity functions (*Pearson Correlation Coefficient* and *Cosine Similarity*) are often a good choice for the quantification of similarity between m/z-images. This seems to be especially valid for samples with high to medium regularity. Interestingly, this result is in line with the recommendations given from the work of Ovchinnikova et al., 2020, in which the use of pre-processing in combination with the *Cosine Similarity* is recommended. However, it was also revealed that the choice of effective similarity functions becomes more difficult with an increasing irregularity of the sample. This result is supported by the observation that alternative image features, such as gradients or magnitudes, as well as the preservation of local neighborhoods seem to gain in importance with increasing irregularity.
The generally low SoRC scores of the global methods indicate that discarding any positional relationship between the pixels is likely to have a negative impact on the clustering result. These observations confirm the initial assumption that the choice

of the similarity function should be carefully considered and tested, especially for samples with low regularity.

Another observation showed that the selected cluster indices (SCS and CHI) evaluate the clustering quality more often differently with increasing irregularity of the sample. This observation indicates that the inherent cluster structures for samples with low regularity are more ambiguous than for samples with high regularity, i.e. the data distribution for samples with low regularity exhibits less pronounced cluster structures.

Finally and probably the most important observation is that there is no pipeline setup that can be universally recommended. The presented results indicate that it is recommended to build a repertoire of different functions and methods which can be evaluated repeatedly for each new data set or analysis objective. Workflows such as the SoRC workflow provide an automatable solution for such repeated evaluation and provide easily accessible and visually presentable results with suggestions for pipeline setups. This saves a time-consuming execution, as well as a tedious pairwise comparison and evaluation of each pipeline setup and its parameters individually. Besides the evaluation of different pipeline setups, workflows such as the SoRC workflow can also be used to evaluate different parameters for the individual methods. However, the more variations are tested, the more computation time is needed.

If time or computational resources are limited it is advisable to use either the *Pearson Correlation Coefficient* or the *Cosine Similarity* function in combination with basic pre-processing procedures and the integration of a small scale-space representation. This recommendation is also supported by the work of Ovchinnikova et al., 2020. Chances are high that such a setup provides good results for regularly structured samples and at least acceptable results for more irregularly structured samples.

All results taken together, it becomes clear that streamlining the comparison of pipeline setups through evaluation workflows with numerical quality estimators is a great support to increase the quality of analysis pipelines and to improve the efficiency of the evaluation process.

4.2.9 Improvements and Future Research

The SoRC workflow in its present form provides only a prototype to demonstrate the importance of evaluating different pipeline setups. Currently, the SoRC workflow can only be used to evaluate clustering as an analysis objective and its execution

requires some programming knowledge. However, the underlying principle of the presented methodology can be extended to any analysis objective, as long as one can define at least one numerical value to estimate the quality of a result. Therefore, a future research topic could be the identification or development of quality indices for further analysis objectives. This way, the presented workflow scheme could easily be adapted for other analysis tasks.

A detailed comparison of the cluster composition between different pipeline setups with high or even identical SoRC values showed that the compositions are not necessarily the same. The reason for this is probably that each pipeline setup, including the clustering method, has unique strengths and weaknesses. For this reason, we believe that it might be worth to explore the idea of computing consensus clusters for a selection of pipeline setups with a high SoRC score, either within the same or even between different clustering methods. In this way, the different setups might complement each other to compensate for some of their individual weaknesses, which could lead to optimized clustering decisions.

Another possible improvement could be to include more cluster indices that are not strictly correlated to increase confidence in the majority voting. A detailed examination of the m/z-image compositions of different clusters showed that some SoRC scores are prone to single "junk clusters". A "junk cluster" describes a cluster in which all ambiguous data points are assembled, which can lead to the artificial improvement of a cluster index. These "junk clusters" are also usually larger than most of the other clusters. The integration of indices that take such and similar problems into account could further improve the quality estimation process.

Remark The remaining parts of this thesis will proceed with the data sets processed by ProViM as presented in Chapter 2 . We will refrain from using the SoRC workflow and from any recommendations that emerged from the presented results. The reason is to avoid any potential bias that could be introduced by the SoRC workflow suggestions and also to demonstrate the general applicability of the following methods. Since there could be potential for optimization, the results of the following sections and chapters could be analyzed later with the SoRC workflow. However, this will not be the subject of this thesis, since the focus is on the exploratory visual analysis of MSI data.

4.3 Community Detection on m/z-Image Similarity Graphs

The previous section discussed the quantification of similarity between m/z-images in the context of clustering. This section presents a novel methodology to cluster m/z-images according to their spatial distribution. The general idea and early basic implementation of this approach date back to my master thesis. Due to the promising results of this pilot study, the development was continued as part of this thesis. At this stage, the methodology received substantial extensions, re-implementations and optimizations. The resulting full-fledged analysis method is presented in this section.

Graphs and networks are well-known data structures within most biological communities. The reason is that in most of these communities the analysis of metabolic pathways is an important branch of research. A metabolic pathway describes a path through a metabolic network, where the structure of a network can be formally described by a graph. This led to the motivation to use graph-driven clustering approaches to cluster m/z-images. In recent years, graph-driven clustering approaches gained an increased amount of attention within the research field of social network analysis. Due to the strong influence of the social network analysis community, the term community detection became established in place of clustering. Therefore, this section will retain this terminology as long as the clustering is applied to a graph, i.e. the term cluster becomes community and clustering becomes community detection.

To apply algorithms for community detection, the given data must be transformed to a graph. However, the data of the spatial MSI domain are provided as a collection of individual m/z-images. Therefore, a transformation is required, which projects the collection of m/z-images onto a graph structure. An outline of the procedure developed for this purpose is illustrated in Figure 4.7. Details of the individual steps are discussed below.

4.3.1 Building the m/z-Image Similarity Graph

In the following, some definitions are given for terms used throughout this chapter.

Graph: A graph is defined as a tuple $G = (V, E)$, where $V = \{v_0 \ldots, v_{|V|-1}\}$ is the set of elements, which are referred to as vertices, and $E = \{(v_i, v_j), (v_{i'}, v_{j'}), \ldots\}$ is

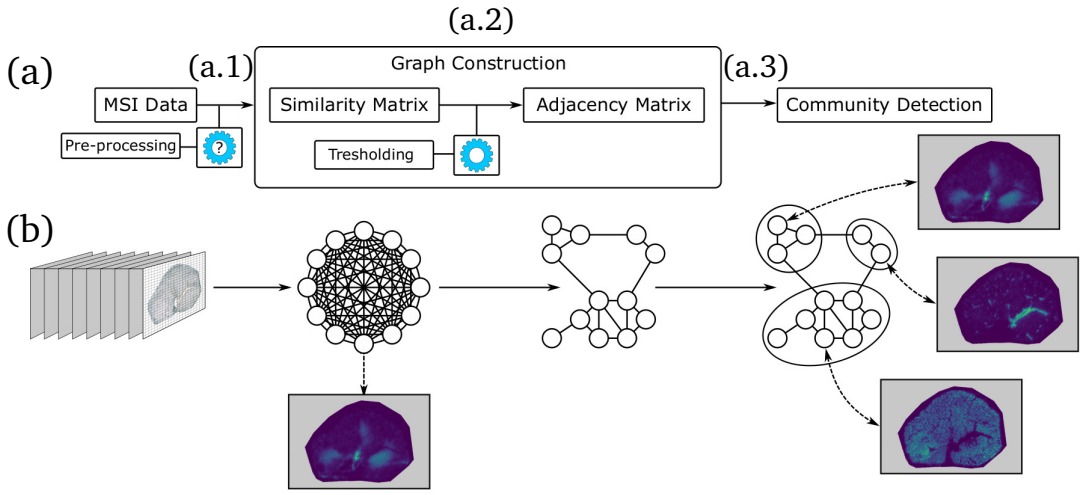


Fig. 4.7.: Outline of the procedure to apply community detection on a collection of m/z-images – (a) Illustration of the procedure as flow chart diagram. (b) Graphical illustration of the data structures resulting from (a). (a.1) After optional pre-processing, the pairwise similarities between the m/z-images are computed and stored as a similarity matrix. This similarity matrix corresponds to a fully connected graph, where each vertex corresponds to an m/z image and each edge corresponds to a similarity value. (a.2) A threshold method is applied to transform the similarity matrix into an adjacency matrix. This adjacency matrix corresponds to the final graph structure to which methods for community detection can be applied. (a.3)The application of community detection methods leads to groups of similar m/z-images.

a set of tuples of vertices. The total number of all vertices and edges is defined by $|V|$ and $|E|$, respectively.

Undirected graph: An undirected graph is defined as a tuple $G = (V, E)$, where V is a set of vertices, and $E = \{\{v_i, v_j\}, \{v_{i'}, v_{j'}\}, \dots\}$ defines a set of unordered tuples of vertices. These unordered tuples are also referred to as two-sets.

Weighted graph: A weighted graph is a graph for which a mapping function assigns numerical values to each of the vertices in V ($v_i \to \mathbb{R}$), or each edge in E ($e_j \to \mathbb{R}$) or both. A graph with weights on its vertices is called vertex-weighted and with weights on its edges is called edge-weighted.

Self-loops: A self-loop defines an edge that connects a vertex with itself $E_j = (v_i, v_i)$. A graph without self-loops is called loopless (or loop-free).

Path: A path $\mathfrak{p}$ is defined as a series of vertices $\mathfrak{p} = (v_0, \dots, v_i, v_{i+1}, \dots, v_{|\mathfrak{p}|-1})$ of size $|\mathfrak{p}|$, where each pair of vertices v_i and v_{i+1} are connected by an edge. The length of a path is defined by the total number of edges in that path $|\mathfrak{p}| - 1$. The length of a path is also referred to as the number of "hops". For weighted graphs, the length

of a weighted path can be defined by the sum of weights of the vertices, edges or both.

Connectivity: Two vertices v and v' are connected if there exists a path that starts with v and ends with v'. A graph is connected if there exists at least one path between every pair of vertices. Otherwise, the graph is called disconnected.

Connected components: For a disconnected graph G, a connected component refers to each sub-graph $g_i \subset G$ that is connected.

Degree: The degree of a vertex of an unweighted graph is defined as the number of all attached edges. For a weighted graph, the degree of a vertex is defined as the sum of the weights of all attached edges.

Clique: A clique is defined as a set of nodes V' that is fully connected $E' = V' \times V'$.

Network: In this thesis, we assume a theoretical differentiation between the terms network and graph. A graph is a mathematical model and a data structure as described above. A network describes the formalization of relationships between entities. A graph-based network describes a data structure, where the entities are represented by vertices and their relationships by edges.

The m/z-Image Similarity Graphs

An m/z-image similarity graph (MISG) is a graph-based network, where each vertex represents an m/z-image and each edge represents the similarity between two m/z-images.

The construction of a MISG is a two-step process. First, a similarity function $\delta(\mathcal{I}_z, \mathcal{I}_{z'})$ is applied to quantify the similarity between every pair of m/z-images. The result is a similarity matrix $\mathcal{S} \in \mathbb{R}^{Z \times Z}$, where Z is the total number of m/z-images. As illustrated in Figure 4.7, $\mathcal{S}$ can be interpreted as a fully connected graph. However, to apply community detection methods effectively, a much sparser graph is required. So the second step is to transform the similarity matrix into a much sparser adjacency matrix $\mathcal{S} \rightarrow \mathcal{A}$. This adjacency matrix defines the structure of the final MISG.

This transformation $\mathcal{S} \rightarrow \mathcal{A}$ is based on the assumption that there exists a threshold value γ, which represents a lower limit for biologically relevant relationships. This means that any edge with a weight below γ is unlikely to represent a biologically relevant similarity. The definition of such a threshold γ in an automated way is a no trivial task. Different approaches to tackle this problem are discussed in Section 4.3.1. After γ has been computed, the transformation $\mathcal{S} \rightarrow \mathcal{A}$ can be applied according

to Equation (4.28) to obtain a weighted MISG or according to Equation (4.29) to obtain an unweighted (binary) MISG. The choice of which transformation to use depends on further use.

$$\mathcal{A}_{i,j} = \begin{cases} 0, & \text{if } \mathcal{S}_{i,j} < \gamma \\ \mathcal{S}_{i,j}, & \text{otherwise} \end{cases} \tag{4.28}$$

$$\mathcal{A}_{i,j} = \begin{cases} 0, & \text{if } \mathcal{S}_{i,j} < \gamma \\ 1, & \text{otherwise} \end{cases} \tag{4.29}$$

Consequently, we provide the following definition: A MISG is defined as a graph $G = (V, E)$, where each vertex $v_i \in V$ represent m/z-image $\mathcal{I}_{z_i}$ and each edge $e_i \in E$ represents the similarity between two m/z-images $\delta(v, v') \in \mathbb{R}$, if $\delta(v, v') > \gamma$. For a binary MISG $\delta(v, v') \in \{0, 1\}$, i.e. the actual similarity value is discarded and an edge only indicates the presence of a similarity above γ. Due to this definition, each vertex can be labeled with the m/z-value of the m/z-image it represents.

In summary, a MISG is an undirected, loopless and weighted or unweighted graph-based network that describes the similarity between molecular distributions of m/z-images above a specified value. For the sake of simplicity, only the term graph will be used in the following, even though the term actually refers to a graph-based network in the context of MISGs.

Edge Reduction

The term edge reduction denotes the transformation $\mathcal{S} \rightarrow \mathcal{A}$, because this transformation is exactly that, the removal of all edges with a weight below a specified value γ. The set of edges that remain after the edge reduction, i.e. the structure of the graph, has a major impact on the community detection result. If there are too many edges, no communities will form and if there are too few edges, the communities will disintegrate.

The most naive way to select a value for γ would be to test a large number of arbitrary values. This would involve creating a graph for each γ, performing a community detection, manually analyzing the results and repeating this process if the result is not satisfying. A result can be described as satisfactory if the detected communities provide information or insights into the organization of the feature space of the m/z-images, e.g. by indicating morphological structures or metabolic interactions.

This approach will quickly become unfeasible. To address this problem, we have developed four approximations to propose values for γ. These approximations cannot guarantee that they provide values for γ that lead to the best possible graph for grouping m/z-images into communities. However, they return values for γ, which most likely lead to graphs containing at least some kind of community structures. That means, there will be groups of vertices that are densely connected, indicating structurally similar m/z-images.

Edge reduction by similarity value statistics (ER-STAT)
The edge reduction method ER-STAT uses basic statistical parameters to approximate γ. The approximation is almost identical to the method presented in Section 4.2.7 to determine the number of clusters for hierarchical clustering.

$$\gamma = \mu(\mathcal{S}') + (c\ \sigma(\mathcal{S}')) \tag{4.30}$$

where S' is the set of pairwise similarity values of all m/z-images $\binom{Z}{2}$, $\mu(\cdot)$ is a function for central tendency, which in this case is the arithmetic mean, $\sigma(\cdot)$ is a function for variation, which in this case is the standard deviation, and c is a tuning factor.

The motivation for this method is to define an interval that describes the expected degree of similarity. Any similarity value above this interval is interpreted as "unexpectedly high", which corresponds to the meaning of biologically relevant in this context. The total interval is defined by the expected value $\pm$ the expected variation. A measure of central tendency is used to approximate the expected value, which is in this case is the arithmetic mean, and a measure of variation is used to approximate the expected variation, which in this case is the standard deviation. However, for the edge reduction, only the upper limit of the interval is relevant, which leads to Equation (4.30). By adding a tuning factor c, the definition of biological relevance can be relaxed or tensed. This ER-STAT is also the method used in Section 4.2.7.

Edge reduction by graph statistics (ER-ACC, ER-SUM, ER-PCA)
The edge reduction methods ER-ACC, ER-SUM and ER-PCA are inspired by previous works on biological network analysis [46, 150, 22]. The idea is to generate several candidate graphs by using different values for γ. Then, different properties of the graph are evaluated by graph statistics. In this thesis, such quantifiable graph properties are referred to as quantitative graph properties (QGP). Based on the evaluation, one of the candidate graphs is algorithmically selected for subsequent community detection. The selected graph yields the value for γ.
For all three edge reduction methods (ER-ACC, ER-SUM, ER-PCA) the computation of γ starts with the determination of a set of candidate thresholds ($\gamma_i \in \Gamma$). The

candidate thresholds in Γ are in descending order and successive thresholds have a constant distance. This constant distance is referred to as step size γ_Δ.

$$\begin{aligned} \Gamma &= (\gamma_{\text{min}}, \ldots, \gamma_{\text{max}}) \\ \gamma_\Delta &= \gamma_{i+1} - \gamma_i \end{aligned} \tag{4.31}$$

Without any prior knowledge, the minimum and maximum thresholds γ_{min} and γ_{max} can be determined by the lowest and highest similarity value:

$$\begin{aligned} \gamma_{\text{min}} &= \min_{(i,j)} \mathcal{S}_{i,j} \\ \gamma_{\text{max}} &= \max_{(i,j)} \mathcal{S}_{i,j} \end{aligned} \tag{4.32}$$

This is because for thresholds below the lowest similarity value no edge will be removed and for thresholds above the highest similarity value no edge remains that could be removed. However, if prior knowledge is available, a narrower interval for γ_{min} and γ_{max} can be selected to reduce computation time.
Three QGPs are relevant for the following approximations to determine γ:

1. The total number of edges $|E|$

2. The average clustering coefficient Λ_c [129]

3. The average efficiency Λ_e [59].

Before the three approximation methods are discussed in detail, the average clustering coefficient and average efficiency are briefly explained, along with the small world principle, which is also important for the motivation of the approximations.

Average clustering coefficient The average clustering coefficient Λ_c quantifies the average tendency of the vertices of a graph to form communities. Λ_c is calculated as the arithmetic mean of the local clustering coefficients of all vertices within a graph (see Equation (4.33)). The local clustering coefficient $\Lambda_{c'}$ of a single vertex quantifies how close a vertex and its immediate neighbors are to a clique. $\Lambda_{c'}$ is calculated as the ratio of the number of edges within the local neighborhood $\mathfrak{V}_i$ of a vertex v_i to the number of all possible edges within this neighborhood (see

Equation (4.34)).

$$\Lambda_c(G) = \frac{1}{|V|} \sum_{v_i \in V} \Lambda_{c'}(v_i) \tag{4.33}$$

$$\Lambda_{c'}(v_i) = \begin{cases} 0, & |\mathfrak{V}_i| \in \{0, 1\} \\ \frac{2|\{e_{q,m} | v_q, v_m \in \mathfrak{V}_i \ \wedge \ e_{q,m} \in E\}|}{|\mathfrak{V}_i|(|\mathfrak{V}_i|-1)}, & |\mathfrak{V}_i| > 1 \end{cases} \tag{4.34}$$

$$\mathfrak{V}_i = \{v_q | e_{i,q} \in E \vee e_{q,i} \in E\}$$

where $|\mathfrak{V}_i|$ is the number of all vertices within the neighborhood of $\mathfrak{V}_i$ and $e_{i,q}$ refers to the edge that connects the two vertices $\{v_i, v_q\}$. Furthermore, $e_{i,q}$ and $e_{q,i}$ are identical for undirected graphs.

Average efficiency The average efficiency quantifies the efficiency of information exchange between vertices within a graph. The information exchange becomes more efficient the shorter the paths between each pair of vertices become. Thus, the average efficiency is defined as the reciprocal of the average shortest path length (see Equation (4.35)).

$$\Lambda_e(G) = \frac{1}{|V|(|V|-1)} \sum_{\substack{v_q \neq v_m \\ v_q, v_m \in V}} \frac{1}{|\mathfrak{p}(q, m)|} \tag{4.35}$$

where $|\mathfrak{p}(q, m)|$ is the length of the shortest path between two vertices.

Small-world graph A small-world graph is characterized by two main structural features:

1. Most vertices of the graph are not neighbors of each other, but the neighbors of any given vertex are likely to be neighbors of each other, i.e. existing neighborhoods show a high degree of connectivity and a high potential to form communities. Properties that describe this structural feature are referred to as segregation properties and measures that quantify this structural feature are referred to as segregation measures.

2. Most vertices can be reached from most vertices by short paths. Properties that describe this structural feature are referred to as integration properties and measures that quantify this structural feature are referred to as integration measures.

In general, small-world graphs are defined as an intermediate between regular and random graphs [129].

Back to edge reduction by graph statistics
In Zahoránszky-Kőhalmi et al., 2016 the authors proposed to compute the average

clustering coefficients (Λ_c) for a set of candidate graphs, which are generated from a set of thresholds Γ. In their work, the value for γ is determined by the threshold that is associated with the first local maximum of Λ_c after an initial decrease.
The motivation for using the average clustering coefficient is based on the observation that adding edges to a graph increases its overall connectivity, but not necessarily the value of Λ_c. This effect occurs when a new edge adds a vertex to a neighborhood that is not or only sparsely connected to that neighborhood. The result is a decrease of the local clustering coefficients, which translates into a lower average clustering coefficient. An illustration of this effect is shown in Figure 4.8. Conversely, this means that the removal of edges can lead to an increase of Λ_c. It follows that a monotonic decrease of the edges does not necessarily cause a monotonic decrease of the average clustering coefficient. Thresholds that lead to an increase of Λ_c, even though edges are removed, indicate states of the graph in which the formation of communities is likely to increase.

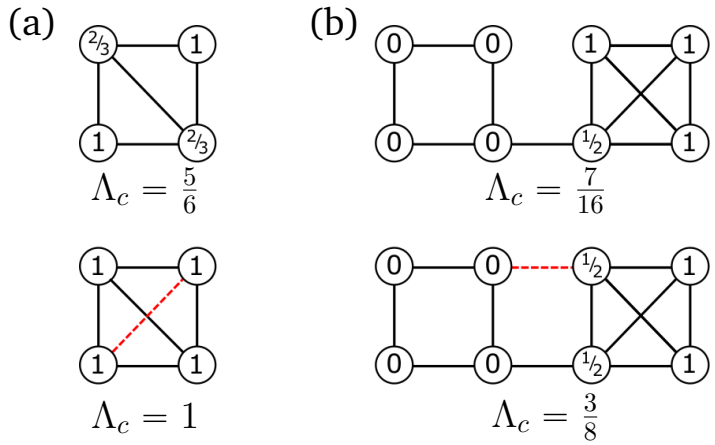


Fig. 4.8.: Illustration of a monotone and a non-monotone behavior of the average clustering coefficient after the addition of an edge – The addition of an edge to a given graph can result in either an increase (a) or decrease (b) of the average clustering coefficient (Λ_c). The numbers in each vertex show the respective local clustering coefficients ($\Lambda_{c'}$). (a) The addition of the edge (dashed red line) increases Λ_c from $\Lambda_c = 0.8\overline{3}$ to $\Lambda_c = 1$. (b) The addition of the edge (dashed red line) decreases Λ_c from $\Lambda_c = 0.4375$ to $\Lambda_c = 0.375$ (the figure is adapted from Zahoránszky-Kőhalmi et al., 2016, Figure 2).

Several of our experiments with different MISGs revealed that the method presented in Zahoránszky-Kőhalmi et al., 2016 is not stable enough to consistently generate MISGs with an adequate number of edges to form communities, i.e. the results of the community detection methods were not competitive with other, more traditional clustering methods. Due to the given task to detect communities of m/z-images, the desired graph structure can be characterized by two properties:

1. The graph should feature highly connected local neighborhoods (communities), i.e. the graph offers a high potential to form communities.

2. The local neighborhoods (communities) should be connected by a solid number of connections to understand how they relate to each other.

The combination of these two structural properties is the previously introduced small-world property, also known as small-worldness [129].
In order to incorporate the small-world property into the threshold selection, we combine the average clustering coefficient $\Lambda_c \in \mathbb{R}^{|\Gamma|}$ with the average efficiency $\Lambda_e \in \mathbb{R}^{|\Gamma|}$. For a small-world graph both of these measures should be balanced and return large values.
The third QPG that has been included in the approximation of γ is the total number of edges $|E| \in \mathbb{R}^{|\Gamma|}$. The reason is that both the average clustering coefficient Λ_c and the average efficiency Λ_e by their definition correlate strongly with the number of edges.
To identify thresholds where Λ_c and Λ_e yield high values independent of the total number of edges $|E|$, the three sets of values (Λ_c, Λ_e, $|E|$) are first normalized to a common value range between $[0, 1]$ according to Equation (4.4).

$$\Lambda_c \mapsto [0,1], \quad \Lambda_e \mapsto [0,1], \quad |E| \mapsto [0,1] \tag{4.36}$$

Then, $|E|$ is used as a baseline and is subtracted from Λ_c and Λ_e. The result of this baseline subtraction are denoted as $\hat{\Lambda}_c$ and $\hat{\Lambda}_e$. The relationship between Λ_c, Λ_e and $|E|$ is illustrated in Figure 4.9. It can be seen that the subtraction of $|E|$ leads to pronounced maxima in $\hat{\Lambda}_c$ and $\hat{\Lambda}_e$.

$$\hat{\Lambda}_c = \Lambda_c - |E| \tag{4.37}$$
$$\hat{\Lambda}_e = \Lambda_e - |E| \tag{4.38}$$

With ER-ACC, ER-SUM and ER-PCA, we propose three different methods to approximate the threshold γ based on the presented QGP.

ER-ACC focuses only on the segregation property to focus on graph structures with dense communities. Among all candidate thresholds Γ, the final threshold $\gamma \in \Gamma$

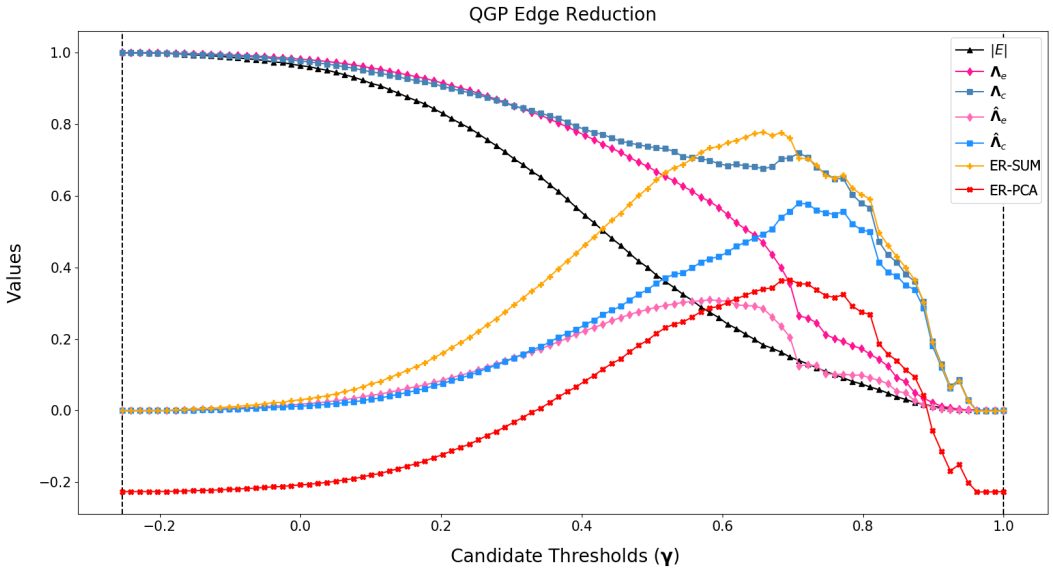


Fig. 4.9.: Illustrative plot that demonstrates the relationship between the individual QGPs used to approximate the edge reduction threshold – The example is computed on the barley seed data set $\mathcal{I}^{\mathrm{B}}$ and the Pearson correlation coefficient as similarity function. The minimum and maximum thresholds are defined by the minimum and maximum similarity value (-0.25363 and 1.0). Both are indicated by vertical lines. Λ_c and Λ_e are already normalized to a value range of $[0, 1]$.

is set to the value that generates the candidate graph that results in the highest baseline-normalized average clustering coefficient $\hat{\Lambda}_c$:

$$\gamma = \underset{\gamma_i \in \Gamma}{\arg\max}\, \hat{\Lambda}_c(G(\gamma_i)) \tag{4.39}$$

ER-SUM and ER-PCA include both the segregation and the integration property to focus on graph structures with dense communities, without neglecting the information provided by the edges between communities. Since the main objective is to find densely connected communities, the segregation property should have a higher priority than the integration property.

ER-SUM sets the final threshold γ to the candidate threshold that generates the candidate graph that yields the highest value for the sum between the baseline-normalized average clustering coefficient $\hat{\Lambda}_c$ and the baseline-normalized average efficiency $\hat{\Lambda}_e$:

$$\gamma = \underset{\gamma_i \in \Gamma}{\arg\max}(\hat{\Lambda}_c(G(\gamma_i)) + \hat{\Lambda}_e(G(\gamma_i))) \tag{4.40}$$

ER-SUM satisfies the priority condition by the observation that $\hat{\Lambda}_c$ generally yields larger values than $\hat{\Lambda}_e$, which gives the segregation measure a stronger influence on the sum.

ER-PCA uses a different approach to set the final threshold γ. We observed that the PCA can be used as an effective weighting function to weight segregation and integration against each other. This follows from the observation that the calculated values for $\hat{\Lambda}_c$ for the different candidate graphs feature a larger variance than the values calculated for $\hat{\Lambda}_e$. Consequently, if the PCA is applied to a two-dimensional sample-feature matrix in which the number of samples is equal to the number of candidate thresholds $|\Gamma|$ and the features are given by $\hat{\Lambda}_c$ and $\hat{\Lambda}_e$, then $\hat{\Lambda}_c$ has a stronger influence within the first principal component than $\hat{\Lambda}_e$ due to the larger variance.

$$\mathcal{Y} = \text{PCA}(\mathcal{Q}) \tag{4.41}$$

$$\gamma = \begin{cases} \arg\min_{i \in \{0,\dots,|\Gamma|-1\}} \mathcal{Y}_{i,0}, & \text{if} \mathcal{Y}_{0,0} > 0 \wedge \mathcal{Y}_{|\Gamma|-1,0} > 0 \\ \arg\max_{i \in \{0,\dots,|\Gamma|-1\}} \mathcal{Y}_{i,0}, & \text{otherwise} \end{cases} \tag{4.42}$$

where $\mathcal{Q} \in \mathbb{R}^{|\Gamma| \times 2}$ is a sample-feature matrix, where the features are given by $\hat{\Lambda}_c(G(\gamma_i))$ and $\hat{\Lambda}_e(G(\gamma_i))$. $\mathcal{Y} \in \mathbb{R}^{|\Gamma| \times 2}$ is the ordered matrix, where each column is an eigenvector (principal components) resulting from the PCA. The order of $\mathcal{Y}$ is descending according to the eigenvalues associated with each eigenvector.

We observed that if the first principal component starts and ends with a value above zero, then it has a defined minimum which determines γ. Otherwise, it has a defined maximum which determines γ.
In many cases, the result of ER-PCA will coincide with ER-ACC. This is because the average efficiency only influences the first principal component if its variation is strong enough.

So the general procedure for all three methods can be summarized as follows:

1. Determination of an ordered set of candidate thresholds with constant distance according to Equations (4.31) and (4.32).
2. Creation of candidate graphs for each candidate threshold $G(\gamma_i)$.
3. Computation of the total number of edges $|E|$, the average clustering coefficient Λ_c according to Equation (4.33) (ER-ACC, ER-SUM, ER-PCA) and the average efficiency Λ_e according to Equation (4.35) (ER-SUM, ER-PCA).
4. Normalization according to Equation (4.36)
5. Baseline-normalization according to Equation (4.37)
6. Computation of the final threshold γ by either ER-ACC (Equation (4.39)), ER-SUM (Equation (4.41)) or ER-PCA (Equation (4.40)).

Remark: The step size γ_Δ can be interpreted as a resolution parameter. A smaller value for γ_Δ requires more computation time, but a large value for γ_Δ might miss well-structured graphs. If prior knowledge is available, a small value for γ_Δ may be compensated by narrowing down the interval $[\gamma_{\text{min}}, \gamma_{\text{max}}]$.

Comparison of the edge reduction methods

Table 4.2 shows a comparison of the four proposed edge reduction methods (ER-STAT, ER-ACC, ER-SUM, ER-PCA). A total of 32 community detection results were computed for this comparison. The four known data sets were combined with two similarity measures to calculate eight similarity matrices. For each combination of data set and similarity function, the four proposed edge reduction methods were used to generate one graph each. Then the *Louvain method* [14] was used to compute communities in these graphs (details on community detection are discussed below). The quality of the communities serves as a measure of the suitability of the edge reduction method. The quality of a community is interpreted similarly to the quality of a cluster discussed in the "clustering quality" paragraph in Section 4.2.5. Since there is no ground truth for the unsupervised detection of communities, the quality of the results can only be estimated. Seven performance indicators are used for this estimation:

1. The Calinski-Harabasz Index (CHI) [17]
2. A modified version of the Calinski-Harabasz Index (CHI2)
3. The Silhouette Coefficient Score (SCS) [100]
4. The Davies-Bouldin Index (DBI) [23]
5. A modified version of the Davies-Bouldin Index (DBI2)
6. The size ratio of the two largest communities (SR)
7. The total number of communities (Nr)

The Calinski-Harabasz Index, Silhouette Coefficient Score and Davies-Bouldin Index are three commonly known metrics (cluster indices) to evaluate the performance of a clustering method. However, our observations showed that the Calinski-Harabasz Index and the Davies-Bouldin Index susceptible to "mega clusters", i.e. cluster results with such clusters are rated better than they should be. The term "mega cluster" describes a very large dispersed cluster. The formation of such clusters can lead to a situation where many small and compact clusters are found at the expense of a very large dispersed cluster. Due to this susceptibility, it can be observed that such a

clustering can achieve good scores for the two mentioned indices, although a major part of the data is assigned to a single very large cluster. To take this problem into account, we propose the two modified index variants (CHI2, DBI2) and add them as additional estimators.

The ratio of the two largest communities (SR) serves as an estimation of the existence of a mega cluster and is used as a penalty factor. A good community detection result maximizes the Calinski-Harabasz Index and minimizes the Davies-Bouldin Index. This means that larger values are better for the Calinski-Harabasz Index, which is why the index is modified by division with the penalty factor (CHI2 $= \frac{\text{CHI}}{\text{SR}}$). For the Davies-Bouldin index, however, smaller values are better, so it is modified by multiplication with the penalty factor (DBI2 $= \text{DBI} \cdot \text{SR}$).

In addition, SR is used as a separate performance indicator to allow an even better assessment of the existence of a potential mega cluster. To indicate a considerable over- or underestimation of the number of communities, the total number of communities (Nr) is added as an additional indicator.

The comparison shows that for the majority of the evaluated approaches both the selected thresholds and the resulting community detection performances are close together. Using colors to highlight the best values for each performance indicator shows that the results vary considerably depending on the data set, similarity function and performance indicator considered. Therefore, no general superiority of one of the methods over the others is apparent. Interestingly, this observation is consistent with the results discussed in Section 4.2.8 and can be traced back to the dependence of context and pipeline setup. Table 4.2 also shows that ER-STAT in one case and ER-SUM in two cases results in the formation of mega clusters (red color). Additionally, ER-SUM shows one case where the number of communities seems to be underestimated (red color). ER-PCA is not always the best performing method, but its performance is close to the best in most test cases and it does not show particularly bad results. Some combinations of data set, similarity function and community detection method show that communities of higher quality can be generated if the average efficiency is excluding and only the average clustering coefficient is used. However, the degree of similarity between the curves of Λ_c and Λ_e can provide an indication of the complexity of the data and the quality of the community detection result.

If computation time is a major concern, note that ER-PCA is the most time consuming, followed by ER-SUM, then ER-ACC and then ER-STAT. This is because PCA is more time consuming than a summation, the computation of the average efficiency is more time consuming than the computation of the average cluster coefficient and the basic statistical approach is the most time-efficient method.

Tab. 4.2.: Edge reduction comparison – Comparison of the four proposed edge reduction methods (ER-ACC, ER-SUM, ER-PCA, ER-STAT) based on the performance indicators computed on the community detection results. Communities are computed using the *Louvain method*. The tables are grouped by data set and similarity function (Pearson: Pearson correlation coefficient; Cosine: cosine similarity). γ shows the selected threshold values. Seven performance indicators are shown, the Calinski-Harabasz Index (CHI) and a modified version (CHI2), the Silhouette Coefficient Score (SCS), the Davies-Bouldin Index (DBI) and a modified version (DBI2), the size ratio of the two largest communities (SR) and the total number of communities (Nr). The arrows indicate whether lower (↓) or higher (↑) numbers are considered better. The best and second-best values for each column are highlighted in green and yellow. Critical values with regard to the cluster size ("mega clusters") or the number of communities (potential over- or underestimation) are highlighted in red.

Barley – Pearson

Transformation	γ	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
ER-ACC	0.70875	3.4736	2.6052	0.3310	1.9039	2.5385	24/11	13
ER-SUM	0.65810	4.6424	1.8570	0.3902	1.9953	4.9881	40/16	8
ER-PCA	0.69609	3.2792	1.9432	0.3466	1.8422	3.1087	27/16	12
ER-STAT	0.67161	3.6235	3.1293	0.3360	2.0498	2.3734	22/19	11

Barley – Cosine

Transformation	γ	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
ER-ACC	0.87666	3.0980	4.6470	0.3609	1.7523	2.6284	15/10	16
ER-SUM	0.85024	4.1450	4.3631	0.3148	1.9381	2.0401	20/19	11
ER-PCA	0.85024	4.1450	4.3631	0.3148	1.9381	2.0401	20/19	11
ER-STAT	0.83678	2.3635	4.4490	0.3517	1.7264	3.2497	32/17	10

Kidney – Pearson

Transformation	γ	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
ER-ACC	0.60215	16.0447	8.4942	0.4951	1.1703	2.2105	34/18	10
ER-SUM	0.58535	16.0765	8.2679	0.4973	1.1656	2.2664	35/18	10
ER-PCA	0.60215	16.0447	8.4942	0.4951	1.1703	2.2105	34/18	10
ER-STAT	0.51166	16.4580	8.7131	0.4791	1.2251	2.3142	34/18	9

Kidney – Cosine

Transformation	γ	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
ER-ACC	0.81510	10.9980	10.9980	0.3400	1.1273	1.1273	22/22	15
ER-SUM	0.81510	10.9980	10.9980	0.3400	1.1273	1.1273	22/22	15
ER-PCA	0.81510	10.9980	10.9980	0.3400	1.1273	1.1273	22/22	15
ER-STAT	0.79069	12.6039	7.0433	0.4000	1.1736	2.1002	34/19	13

Bladder – Pearson

Transformation	γ	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
ER-ACC	0.46611	8.5888	4.5091	0.3228	1.1379	2.1675	40/21	28
ER-SUM	0.04528	18.4179	8.9119	0.3029	1.9285	3.9856	93/45	4
ER-PCA	0.44781	8.7421	5.0510	0.3434	1.1116	1.9240	45/26	23
ER-STAT	0.38452	9.9463	4.4403	0.3531	1.2827	2.8732	56/25	16

Bladder – Cosine

Transformation	γ	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
ER-ACC	0.56513	9.5521	6.2296	0.2381	1.5514	2.3788	46/30	12
ER-SUM	0.55505	10.2506	8.4108	0.2662	1.5689	1.9121	39/32	11
ER-PCA	0.56513	9.5521	6.2296	0.2381	1.5514	2.3788	46/30	12
ER-STAT	0.60158	7.5412	5.1501	0.2429	1.3400	1.9621	41/28	16

Skin – Pearson

Transformation	γ	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
ER-ACC	0.13540	4.1359	3.6189	0.3910	1.5343	1.7535	16/14	7
ER-SUM	0.08348	4.9342	4.9342	0.3690	1.7727	1.7727	17/17	5
ER-PCA	0.13540	4.1359	3.6189	0.3910	1.5343	1.7535	16/14	7
ER-STAT	0.28986	2.8061	2.6057	0.4055	1.3955	1.5028	14/13	10

Skin – Cosine

Transformation	γ	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
ER-ACC	0.31151	2.5779	2.5779	0.3930	1.3426	1.3426	13/13	11
ER-SUM	0.20443	2.8061	2.6057	0.4032	1.3955	1.5028	14/13	10
ER-PCA	0.31151	2.5779	2.5779	0.3930	1.3426	1.3426	13/13	11
ER-STAT	0.33720	2.3930	2.2089	0.3895	1.2929	1.4006	13/12	12

Community Detection

Due to their strong background in social network analysis research, the terms community and community detection have established themselves as synonyms for clustering and clustering. Consequently, community detection is an ill-posed problem, just like clustering, for which there is no precise definition (see Section 1.3). However, similar to clustering, there are statements that should be generally valid to describe the concepts of communities and community detection.

1. The probability that two vertices are connected is higher if they are members of the same community.

2. The connectivity between vertices within the same community is higher than the connectivity to other vertices of other communities.

Despite the influence of social networks analysis community, community detection can of course be used wherever cluster structures in networks need to be analyzed, e.g. in transport networks, telecommunication networks or biological networks.

There are many different algorithms available to detect communities. A first differentiation can be made between overlapping [87, 144, 27] and non-overlapping [57, 147] approaches. In overlapping community detection, a single vertex can be assigned to more than one community. Thus, it can be viewed as a kind of fuzzy clustering approach. In non-overlapping community detection, a single vertex can only belong to one community. The methodology presented in this thesis will focus exclusively on non-overlapping community detection approaches. This is because non-overlapping communities are easier to visualize and easier to interpret, especially for users who are not experienced in graph and network analysis. The current implementation of the presented methodology for community detection on m/z-images allows the use of two different community detection algorithms. The first is referred to as *Leading Eigenvector method* [81] and the second is usually referred to as *Louvain method* or *Multilevel method* [14]. However, in the following, we will focus only on the *Louvain method*, as it has the advantage to detect hierarchical community structures, hence the name *Multilevel method*. Moreover, different comparative studies have shown that the *Louvain method* is one of the best-performing methods in terms of scaling with network size and detection performance [57, 147].

Louvain method The *Louvain method* is a heuristic approach based on modularity optimization. For a graph that is partitioned into communities, the modularity is a scalar value that measures the density of the edges within communities compared

to the edges between communities [82, 81]. The algorithm consists of two phases, which are repeated iteratively. The first phase assigns each vertex to an individual community. Then, for each vertex v_i all neighboring vertices v_q are evaluated whether a gain in modularity can be achieved by removing v_i from its community and adding it to the community of one of its neighbors v_q. If a positive gain in modularity can be achieved, v_i is placed in the community where the gain is maximum (ties are resolved by some breaking rule). If no positive gain can be achieved, v_i remains in its community. This procedure is repeated sequentially for each vertex until there is no gain in modularity for any vertex. If no further gain is available, a local maximum of modularity is reached and the first phase ends. The second phase creates a community graph by representing each community as a single vertex. The edges between the community vertices are weighted by the sum of the weights of all edges that have connected any two vertices of these communities. Both phases are repeated until no additional gain in modularity can be achieved. The "height" or "level" of the hierarchy is defined by the number of completed iterations, where a completed iteration is defined by the end of the second phase. In a sense, this procedure resembles the concept of agglomerative hierarchical clustering.

In order to evaluate the performance of community detection on MISGs, Table 4.3 compares the computed results to other known clustering methods. For the comparison, the four known data sets ($\mathcal{I}^{\mathrm{B}}$, $\mathcal{I}^{\mathrm{K}}$, $\mathcal{I}^{\mathrm{S}}$, $\mathcal{I}^{\mathrm{U}}$) were combined with two similarity functions (Pearson correlation coefficient and cosine similarity), which resulted in eight test cases. The comparison methods are: hierarchical clustering (once with a given number of clusters and once with an automatically detected number of clusters according to Equation (4.27)), k-means (using a given number of clusters and the squared Euclidean distance due to its construction) and affinity propagation (automatic number of clusters). If the number of clusters must be predefined, it was set to the number of detected communities by the *Louvain method*. The results are evaluated using the same seven performance indicators as for the comparison of the edge reduction methods (see Table 4.2).
It can be seen that the community detection approach achieves on average results that are competitive with the other methods. This is especially true if the seemingly very good results of k-means for the Calinski-Harabasz Index and the Davies-Bouldin Index are neglected. These high values in CHI and DBI for k-means always coincide with low values for the Silhouette Coefficient Score, which is an indicator that these values are achieved at the expense of some very bad cluster decisions.
Consequently, the community detection approach seems to be a competitive and effective approach for clustering m/z-images, at least for the presented test cases.

Tab. 4.3.: Community detection performance comparison – Comparison of the community detection methodology (ER-PCA with *Louvain method*) to other clustering methods (* indicates the use of Equation (4.27) to estimate the number of clusters). The tables are grouped by data set and similarity function (Pearson: Pearson correlation coefficient; Cosine: cosine similarity). Seven performance indicators are shown, the Calinski-Harabasz Index (CHI) and a modified version (CHI2), the Silhouette Coefficient Score (SCS), the Davies-Bouldin Index (DBI) and a modified version (DBI2), the size ratio of the two largest communities (SR) and the total number of communities (Nr). The arrows indicate whether lower (↓) or higher (↑) numbers are considered better. The best and second-best values for each column are highlighted in green and yellow. Critical values with regard to the cluster size ("mega clusters") are highlighted in red.

Barley – Pearson

Transformation	CHI ↑	CHI2 ↑	SS ↑	DBI ↓	DBI2 ↓	SR	Nr
Community Detection	3.2792	1.9432	0.3466	1.8422	3.1087	27/16	12
Hierarchical Clustering	2.7866	0.8066	0.3423	1.8840	6.5083	38/11	12
k-Means	62.0078	15.0322	−0.2592	0.6366	2.6260	66/16	12
Hierarchical Clustering*	2.3229	0.6724	0.3399	1.7173	5.9324	38/11	14
Affinity Propagation	5.0372	3.7779	0.3396	1.7801	2.3735	16/12	12

Barley – Cosine

Transformation	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
Community Detection	4.3631	4.145	0.3148	1.9381	2.0401	20/19	11
Hierarchical Clustering	5.1792	5.0066	0.3212	1.7262	1.7857	30/29	11
k-Means	62.6679	16.6262	−0.1229	0.9546	3.5981	49/13	11
Hierarchical Clustering*	4.2748	1.4249	0.3489	1.8702	5.6105	30/10	13
Affinity Propagation	5.3784	3.8844	0.3726	1.9439	2.6916	18/13	12

Kidney – Pearson

Transformation	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
Community Detection	16.0447	8.4942	0.4951	1.1703	2.2105	34/18	10
Hierarchical Clustering	15.6624	8.5431	0.4828	1.1518	2.1116	33/18	10
k-Means	24.5756	19.2331	0.0537	1.2820	1.6381	23/18	10
Hierarchical Clustering*	13.1337	7.8802	0.4486	1.1112	1.852	30/18	13
Affinity Propagation	17.2066	15.3954	0.4684	1.5327	1.7131	19/17	10

Kidney – Cosine

Transformation	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
Community Detection	10.9980	10.998	0.3400	1.1273	1.1273	22/22	15
Hierarchical Clustering	11.6540	2.4279	0.4558	1.0296	4.9418	48/10	15
k-Means	21.8074	16.8511	−0.0503	1.0460	1.3536	22/17	15
Hierarchical Clustering*	10.7988	3.1652	0.4321	0.9472	3.2315	58/17	11
Affinity Propagation	16.4839	6.2372	0.5004	1.3193	3.4868	37/14	11

Bladder – Pearson

Transformation	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
Community Detection	8.7421	5.0510	0.3434	1.1116	1.9240	45/26	23
Hierarchical Clustering	8.3607	4.0950	0.3168	1.1117	2.2697	49/24	23
k-Means	13.3842	8.0306	0.1640	1.5805	2.6342	20/12	23
Hierarchical Clustering*	8.9552	5.3731	0.3361	1.1045	1.8408	40/24	25
Affinity Propagation	11.3791	6.1059	0.3027	1.4980	2.7918	41/22	19

Bladder – Cosine

Transformation	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
Community Detection	9.5521	6.2296	0.2381	1.5514	2.3788	46/30	12
Hierarchical Clustering	8.6089	2.2547	0.2875	1.2628	4.8217	84/22	12
k-Means	18.7799	14.2468	0.1533	1.7654	2.3272	29/22	12
Hierarchical Clustering*	6.9208	2.6425	0.2969	1.1087	2.9039	55/21	24
Affinity Propagation	12.011	5.8720	0.2801	1.6019	3.2766	45/22	15

Skin – Pearson

Transformation	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
Community Detection	4.1359	3.6189	0.3910	1.5343	1.7535	16/14	7
Hierarchical Clustering	2.7676	2.0393	0.3516	1.6689	2.2649	19/14	7
k-Means	19.1654	2.1903	0.0568	0.7240	6.3354	35/4	7
Hierarchical Clustering*	2.6851	2.4786	0.4031	1.3312	1.4421	13/12	11
Affinity Propagation	6.3821	5.0145	0.3565	1.4757	1.8782	14/11	7

Skin – Cosine

Transformation	CHI ↑	CHI2 ↑	SCS ↑	DBI ↓	DBI2 ↓	SR	Nr
Community Detection	2.5779	2.5779	0.3930	1.3426	1.3426	13/13	11
Hierarchical Clustering	2.6851	2.4786	0.4030	1.3312	1.4421	13/12	11
k-Means	22.5379	5.8794	0.0848	0.6950	2.6640	23/6	11
Hierarchical Clustering*	2.6851	2.4786	0.4030	1.3312	1.4421	13/12	11
Affinity Propagation	6.6550	3.3275	0.3058	1.3572	2.7144	14/7	8

4.3.2 Interactive Visual Exploration of m/z-Image Similarity Graphs

To allow an interactive visual exploration of m/z-image similarity graphs and the analysis of community structures, we have developed a visual analysis (web)tool, called GRINE (analysis of **gr**aph mapped **i**mage data **ne**tworks). The exploratory possibilities offered by an interactive visual tool allow the user to exploit the full potential of the graph structure.

The first version of the GRINE frontend and the community detection offline component (described above) were developed during my master thesis. As part of this thesis, GRINE was functionally extended and its visual coherence was improved. The community detection offline component was overhauled, improved and extended. Later, the second version of GRINE was developed and the name changed to COBI-GRINE (analysis, **c**luster **o**ptimization and **b**iological **i**nterpretation of **gr**aph mapped **i**mage data **ne**tworks). To keep the name short, COBI-GRINE continues to be abbreviated as GRINE in this thesis. The implementation was done in cooperation with two student projects (Kim Wüstkamp and Daniel Göbel) and one bachelor thesis (Daniel Göbel). Under my supervision, Kim Wüstkamp implemented parts of the frontend and minor parts of the backend. Daniel Göbel implemented major parts of the backend and major parts of the frontend. Conception, development and design, minor parts of the implementation in frontend and backend, as well as the implementation of all offline components were done by me.

The interface

According to our design guidelines proposed in Section 1.4 the user interface of GRINE is kept clear and simplistic (see Figure 4.10). It consists of three main visual building blocks which are referred to as displays:

1. Graph display (Figure 4.10c)
2. Image display (Figure 4.10e)
3. List display (Figure 4.10f)

All three displays are interconnected and most actions in one of the displays trigger reactions in the others.
Options that either do not need to be changed frequently during data exploration or are associated with very specific tasks are hidden and can be accessed via the options menu (Figure 4.10a). Functionalities that are frequently needed during data

exploration or cause permanent changes to the graph are accessible via fast-access controls (Figure 4.10b, Figure 4.10d).

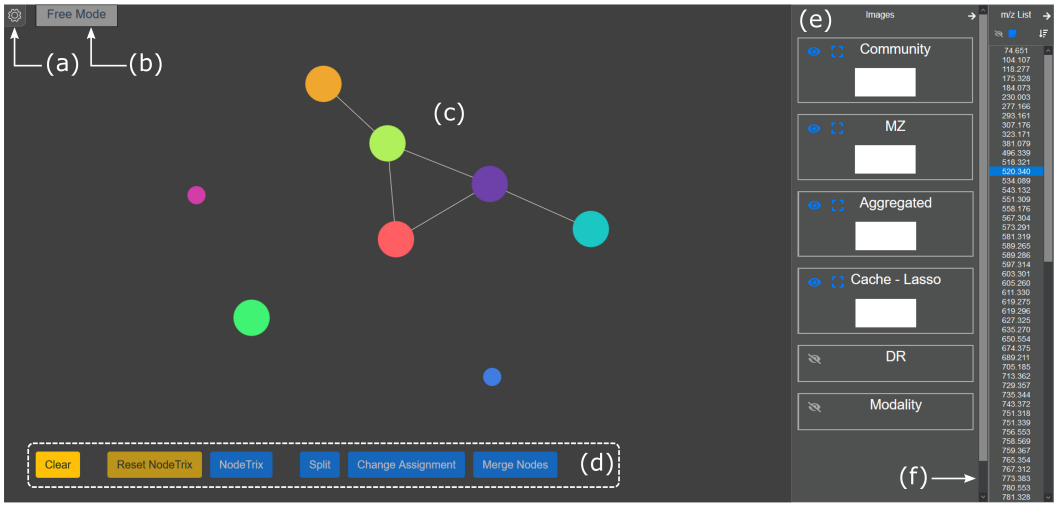


Fig. 4.10.: Overview of the GRINE user interface – (a) Options menu; (b) Control to switch between navigation mode and selection mode; (c) graph display; (d) fast-access controls for graph manipulation; (e) image display; (f) list display with active m/z-list and inactive QGP-list.

The graph display

The graph display visualizes the hierarchical m/z-image similarity graph (MISG). The graph contains two types of vertices and three types of edges.
The two types of vertices are defined as follows:

1. m/z-image vertices referred to as m/z-vertices, where each vertex corresponds to a single m/z-image.
2. Community vertices, where each vertex corresponds to a community of m/z-images.

If a hierarchical algorithm has been used for community detection, the community vertices may have several levels. In fact, a MISG can always be interpreted as a hierarchical graph, in which the m/z-vertices form the lowest hierarchy, followed by the first level of communities. Depending on the data and the applied community detection method, these communities can also be merged into new communities to create another hierarchical level. GRINE uses two different visual cues for the vertex encoding, which is illustrated in Figure 4.11. Color is used to differentiate between communities. At the highest hierarchical level, each community is assigned to a unique color. This can be described as a categorical colormapping problem with

an uncertain number of classes. To solve this problem, the communities are mapped with a linear constant distance on a rainbow colormap. The other visual cue is size, which is used to distinguish between the different hierarchical levels. The size of a vertex decreases from the highest to the lowest hierarchy. This is intuitive because with an increasing hierarchical level the number of grouped vertices increases as well.
In addition to the two types of vertices, there are three types of edges:

1. m/z-image edges referred to as m/z-edges, which represent a connection between two m/z-vertices.

2. Community edges, which represent a connection between two communities of the same hierarchy level.

3. Hybrid edges, which represent a connection between two vertices of different hierarchy levels. This means that a vertex of a lower hierarchy has a connection to another vertex of the same hierarchy, which is currently hidden inside of a community of a higher hierarchy. Hidden describes a state in which a vertex is not visually visible because its parent community is already displayed. The principle of hybrid edges is illustrated in Figure 4.11.

m/z-edges and community edges are both encoded with solid lines. The length of these edges is scaled inversely according to the degree of similarity of the connected vertices. The shorter the edge, the higher the similarity. Consequently, both edge classes are primarily responsible for the final graph layout, which is computed by a force-directed layout algorithm (using d3-force [16]). For m/z-edges the term similarity refers to the computed entry in the adjacency matrix. For community edges, the term similarity refers to the arithmetic mean of the similarities of all edges connecting any two nodes of the respective communities. Hybrid edges are encoded with dashed lines. Unlike the other two classes, hybrid edges do not influence the layout. They only serve as a visual indicator for connections to another hierarchical level.

To be able to handle graphs and sub-graphs of different sizes, the graph display offers various interactions. A stepwise zoom function is implemented to switch back and forth between the entire graph and smaller segments. This allows to analyze subsets of vertices and their connections in detail but also enables a quick change to a larger context. To enable intuitive navigation when zoomed, the graph display can be moved freely with a drag and drop interaction.

To compute the position of the vertices on the display a force-directed layout is used. If the vertices have an unfavorable starting position, well-structured subsets

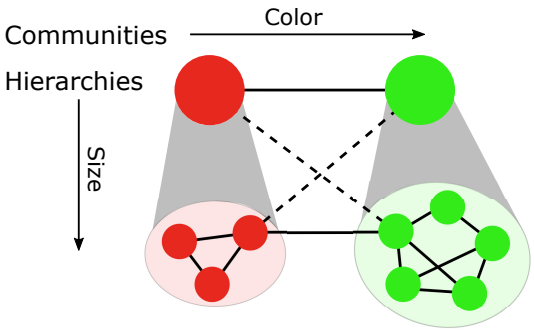


Fig. 4.11.: Visual cues in the MISG encoding – Color is used to distinguish between communities. Size is used to distinguish different levels of hierarchies, including the differentiation between community vertices and m/z-vertices. Edges between vertices of the same hierarchy are solid, while edges between hierarchies (hybrid edges) are dashed.

of vertices can start to become cluttered, or that interesting structures are not visible. To solve this problem, every vertex can be moved freely through a drag and drop interaction. This way the graph layout can be manually reordered, which supports the analysis and presentation of specific structures. To ensure that the edge lengths maintain their meaning, each drag and drop interaction will reactivate the force-layout algorithm to correct the manual reordering. The extent to which the algorithm intervenes in the layout can be adjusted in the options menu.

To be compatible with most community detection methods, GRINE is designed to handle graphs with hierarchically organized communities. This includes for example the output of the *Louvain method* discussed in Section 4.3.1. To minimize the number of visual elements when GRINE is started, the graph is displayed as a community graph at the highest available hierarchical level. The visualization of the entire graph at its lowest hierarchical level can lead to an overloaded display and a cluttered layout even for graphs of small to medium size, e.g. a graph representation of the barley seed data set with 101 vertices and 761 edges already shows cluttered structures, although this is counted as a rather small graph. To address the cluttering problem, each vertex can be folded into a higher hierarchy level or unfolded into a lower hierarchy level. This selective filtering approach enables a flexible alternation between the different hierarchies, allowing a trade-off between the analysis of detailed lower hierarchical structures and visual clarity. After each folding and unfolding operation, edges are automatically added or removed. In addition, the force-layout algorithm is triggered to maintain a proper representation of edge lengths and to improve visual clarity.

As discussed in Section 4.3.1, a MISG is a data structure consisting of vertices and edges, where each vertex represents an m/z-image and each edge represents

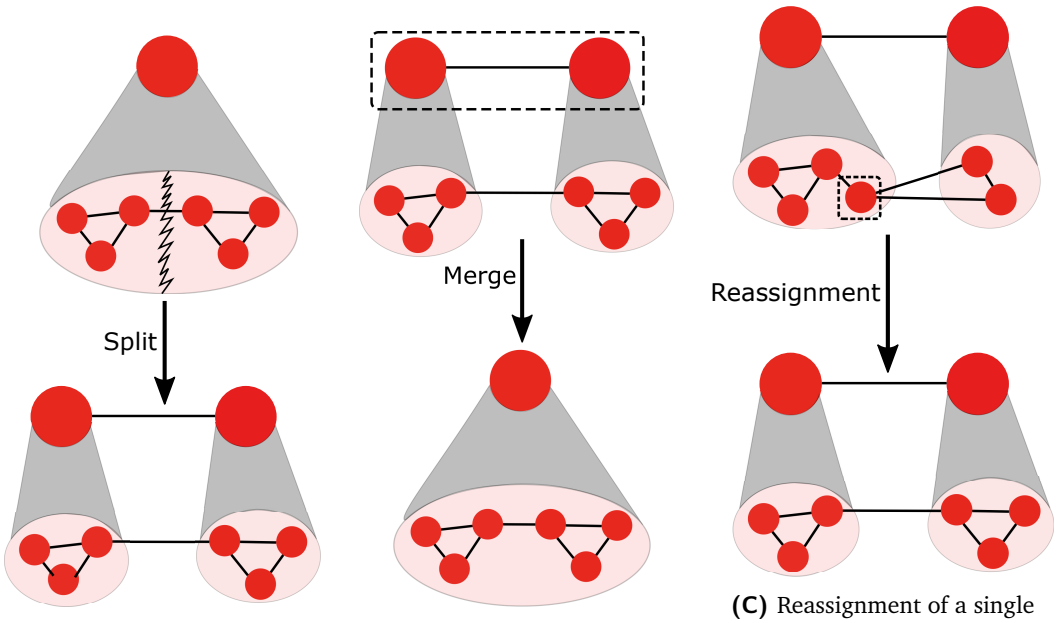
a similarity. In order to analyze a MISG properly, it is necessary to connect the information provided by the graph with the information provided by the m/z-images. To fulfill this requirement the graph display and the image display are closely linked. If a single vertex or a group of vertices is selected, the respective panels of the image display are used to visualize either individual m/z-images or an aggregation of a group of m/z-images, according to the selection. More details about the different images panels follow below.

For a detailed analysis, the custom selection of vertices is a valuable tool, as it enables the exploration of groups of vertices outside the predefined communities. With custom selections, the user can manually search for things like sub-communities, potential community assignment errors or potential community improvements. The ability to make custom selections is implemented through a lasso selection tool. Most users can control a lasso selection tool intuitively because it is known from various other applications and it is quite similar to the rectangular selection tool from desktop environments. To provide persistent visual feedback that shows which vertices are currently selected, the shape of the selected vertices changes from circular to rectangular. The change in shape is accompanied by a pop-up animation to further emphasize the visual feedback of a selection, i.e. that the symbol is briefly enlarged.

As discussed in Section 4.3.1, community detection is a problem that is not precisely defined. Therefore results may differ from expectations, e.g. small communities may not be detected because they have been merged into larger communities or assignments may not match the expectations of the analyst. To enable analysts to use their expertise to change or improve communities, community manipulation features have been added. These features include the splitting and merging of communities as well as a reassignment of the membership of vertices. All three manipulation features are illustrated in Figure 4.12. In combination with the lasso selection, these functions provide a powerful tool to interact with the graph and its communities. The modified graph can be easily exported as a JSON file and later imported again, allowing easy storage and exchange.

Despite the use of a force-directed layout algorithm and the possibility of manual rearrangement, parts of graphs with a high edge density will still tend to be cluttered and impair visual clarity. To facilitate the analysis of these cluttered areas in the overall context of the graph, a variant of the NodeTrix technique [42] was implemented.

NodeTrix represents a selected subset of vertices as an adjacency matrix. This adjacency matrix is visualized as a heatmap, which allows an intuitive and fast analysis



(A) Split of one community into two.

(B) Merge of two communities into one.

(C) Reassignment of a single vertex from one community into another.

Fig. 4.12.: Community manipulation – Illustration of the three community manipulation features.

of the connections (edges). In addition, the adjacency matrix is embedded within the graph structure to preserve the context of the vertices. The implementation in GRINE allows one active NodeTrix at a time. The NodeTrix transformation is a valuable tool to improve the visual clarity of densely connected subsets of vertices. It also provides a fast overview of structural features that could otherwise remain hidden due to clutter. Examples for such structural features are overall strong interconnections (homogeneity), overall weak interconnections (heterogeneity) or strongly interconnected subgroups (sub-communities). Thus, the NodeTrix transformation can be used to evaluate the quality of communities or to identify starting points for community manipulations. Figure 4.13 illustrates the NodeTrix transformation and the three structural feature examples. The NodeTrix transformation can also be used to identify ambiguous community detection decisions that are based on problematic graph structures, e.g. offshoots in communities (see Figure 4.14).

To accelerate the exploration of the MISG, annotation labels are triggered on hover for each vertex or NodeTrix cell. This way, the user is provided with additional information about the focused element without overloading the display.

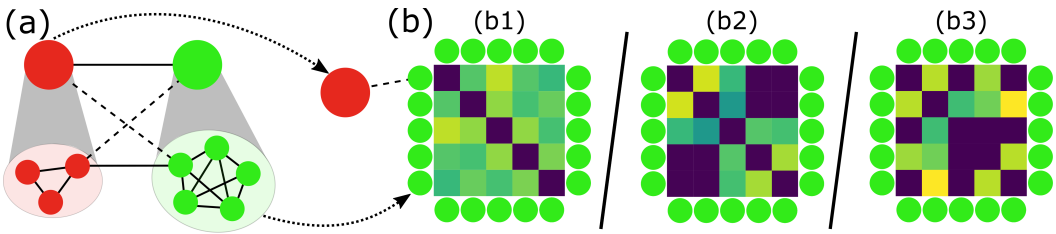


Fig. 4.13.: Example of a NodeTrix transformation – The green community vertex of the graph (a) is transformed into a NodeTrix representation (b). (b1) illustrates a homogeneous NodeTrix example, (b2) illustrates a NodeTrix example with potential sub-communities and (b3) illustrates a heterogeneous NodeTrix example. The example uses "viridis" as colormap to visualize the NodeTrix heatmaps.

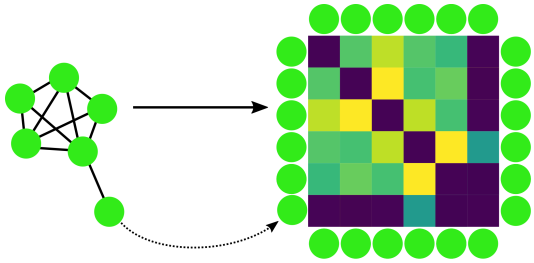


Fig. 4.14.: NodeTrix offshoot example – The graph (left) shows an offshoot structure attached to a clique. The offshoot structure can also be spotted in the respective NodeTrix transformation (right). The dashed arrow indicates the offshoot vertex. "Viridis" is used as colormap to visualize the NodeTrix heatmap.

The image display

The graph display offers three different types of selection:

1. Selection of community vertices
2. Selection of m/z-vertices
3. Custom selections

Since each vertex represents one or more m/z-images, these selections correspond to the selection of one or more m/z-images. For an efficient analysis of a MISG, it is necessary to have direct access to the m/z-images of every vertex selection. Otherwise, the MISG would lose most of its value, since the graph structure is intended to serve as a medium that facilitates the analysis of m/z-image communities. For this reason, the image display and the graph display are closely linked. The image display contains six different panels (see Figure 4.10), which are labeled with: "Community", "MZ", "Aggregated", "Cache – Lasso", "DR" and "Modality". The first three panels are referred to as MISG panels. These panels are linked to the three different types of selections, i.e. the community panel is linked to the selection of

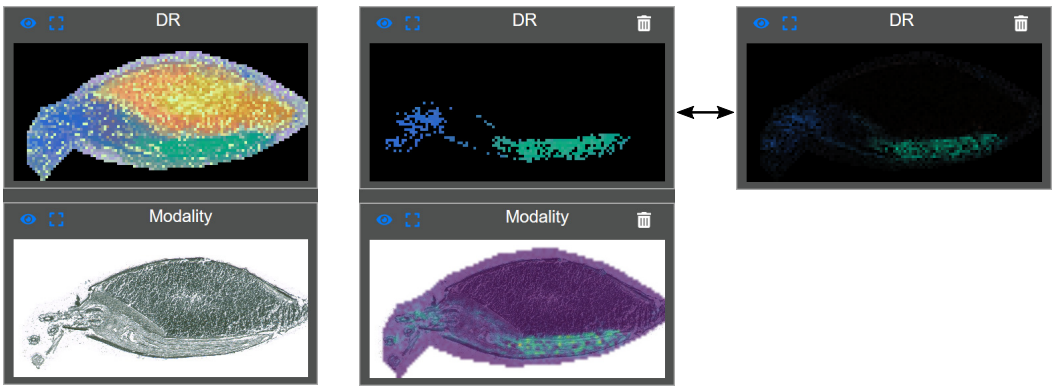
a community vertex, the m/z-panel is linked to the selection of an m/z-vertex and the aggregation panel is linked to custom selections. To examine an m/z-image in the context of its community, selecting a single m/z-vertex does not only show the respective m/z-image, but also the corresponding community image.

The visualization of images for communities and custom selections require the combination of a set of m/z-images into a single image representation. A straightforward way to transform a set of images into a single image is the pixel-wise application of an aggregation function. Examples are the arithmetic mean, median, summation or maximum. However, there are many different aggregation functions available and each of them has a different effect on the final result. Some of these functions also allow different perspectives on the community, e.g. an aggregation with the arithmetic mean represents the average molecular distribution of the community, while an aggregation with the maximum function represents the maximum coverage of a molecular distribution. In order to explore these different perspectives, the function can be adjusted in the options menu. Four functions are available: arithmetic mean, median, minimum or maximum. A similar observation applies to the applied colormap. Different colormaps can emphasize differently structured molecular distributions. This is due to perceptual differences in the mapped color space [118, 21, 98, 97, 40, 136]. To provide more flexibility for the analysis with regard to the color space, a selection of colormaps is available. The selection is intentionally restricted to a set of perceptually uniform sequential colormaps, including “viridis”, “magma”, “inferno” and “plasma” [72]. The reason is that these maps are well suited for visualizing continuous and discrete ordinal data, such as m/z-images. In addition, the minimum and maximum of each image are scaled linearly to a range of $[0, 1]$ before the colormap projection, making use of the entire range of the applied colormap. As a result, the contrast of the molecular distribution patterns is increased and the visual representation is improved.

The panels mentioned so far allow the visualization of any image resulting from a selection within the graph display. However, a possibility to compare molecular distributions is missing. But for detailed analysis, the ability to compare images from any two selections is an important functionality. For example, such a comparison can provide additional information to discover why vertices in the graph are adjacent or far apart. Furthermore, a comparison functionality is especially important to support decisions on community manipulations. Examples would include the comparison of single m/z-vertices to other communities to find potential reassignments or the comparison of two selected subsets of vertices to find sub-communities. To make a comparison possible, every MISG image can be copied (“cached”) into the cache-lasso panel.

Like any technology for measuring biological samples, MSI has its advantages and disadvantages. A major advantage is the ability to measure molecular distributions directly. A major disadvantage is the limited spatial resolution. To compensate for the disadvantages, it can be helpful to complement MSI with other imaging modalities and with the results of other methods. The extension with other modalities and analysis results can be particularly helpful for the analysis and evaluation of community images and custom selections. These images do not represent directly measured molecular distributions and the added context provided by other modalities can facilitate the interpretation of these aggregated distributions. To support the addition of other modalities and analysis results the panels "DR" and "Modality" were added (see Figure 4.15 A). "DR" refers to dimension reduction and presents the RGB projection image (details about the RGB projection image were discussed in Section 3.3). This analysis result is implemented as part of the GRINE offline component. As for VAIDRA, there are twelve dimensional reduction methods to choose from (see Section 3.3). The modality panel allows loading any image file as an additional modality, such as microscopy images or stained images. However, there is no automatic alignment method integrated. This means that the alignment must be performed beforehand, either manually or using other software. To facilitate comparative analysis between the selected MISG images (m/z-image, community image or custom selection image) and the other two image sources, an overlay functionality was implemented, which is illustrated in Figure 4.15. For the modality panel, the overlay is rendered by drawing the MISG image with an adjustable alpha channel on top of the modality image. Since the alpha channel is adjustable, both images can be compared by dynamically decreasing and increasing the opacity (see Figure 4.15 B). For the DR panel, the overlay is computed by a projection of the MISG image into the alpha channel of the DR image. This overlay has two different modes: relative and absolute. In absolute mode, the alpha value is set to one for each pixel that has an intensity value above a selected threshold in the MISG image. The alpha value for all other pixels is set to zero. Technically, this mode uses the overlay as a kind of binary mask that makes overlapping areas visible. In relative mode, the intensity values of the MISG image are directly converted into alpha values. As a result, the DR image is better visible in high-intensity areas than in low-intensity areas (see Figure 4.15 B). As an additional feature, the DR image can also be superimposed with the modality image, which works in the same way that the MISG image is superimposed with the modality image.

The features and functionalities described above are either centered around the image display itself or represent a one-way workflow between the graph display and the image display (graph display → image display). However, for the analysis



(A) UMAP based RGB projection image (top) and brightfield image (bottom).

(B) Overlay of a MISG image with an RGB projection image in absolute mode (top left, threshold equals 20%), relative mode (top right) and an overlay with a brightfield image (bottom, alpha equals 70%).

(C) Overlay of an RGB projection image with a brightfield image (alpha equals 60%).

(D) Overlay of a MISG image with an RGB projection image in absolute mode (threshold equals 20%) that is superimposed with a brightfield image (alpha equals 70%).

Fig. 4.15.: Examples for the overlay functionality – The examples shown are based on the barley data set $\mathcal{I}^{\mathrm{B}}$.

of spatial molecular distributions it is desirable to offer an additional workflow that allows exploring the graph using distribution patterns (image display → graph display). This is especially if further modalities are available. Such a workflow should allow the selection of distribution patterns on any type of image, followed by visually highlighting m/z-vertices or community vertices with similar distribution patterns. An example that uses the workflow image display → graph display could be described as follows: Suppose that the starting point of the analysis is a specific m/z-image or a specific distribution pattern in a histopathologically stained image. In this case, highlighting nodes with distribution patterns that are similar to a selected distribution pattern provides a much more efficient starting point for exploring the graph than a random search for vertices with the desired distribution pattern. We refer to this workflow direction as image-guided search. The principle of the image-guided search is illustrated in Figure 4.16).

GRINE implements the image-guided search in two different ways, depending on whether a MISG image or one of the other two modalities (DR image or modality image) is used. This is because in most cases the value of a pixel in these modalities

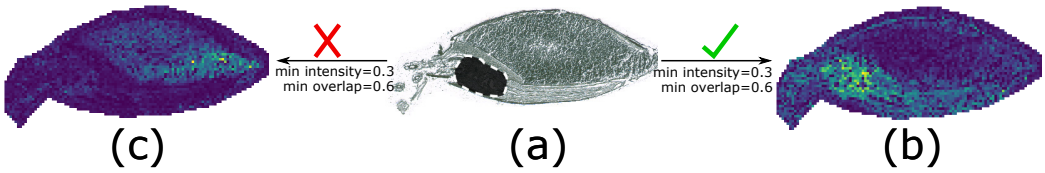


Fig. 4.16.: Example of the image-guided search using the lasso selection tool – The image-guided search is used on the brightfield image of the barley seed. (a) brightfield image of the barley seed, with a lasso selection shown in dark gray. The parameters are set to: minimum considered intensity $= 0.3$ and the minimum required overlap $= 0.6$. This means that each vertex is highlighted whose intensity image has at least 80% active pixels in the selected area. A pixel is considered as active if the intensity value is at least equal to 20% of the maximum intensity of the intensity image. (b) A MISG image that satisfies this condition. (c) A MISG image that does not satisfy this condition.

will have a different interpretation than in the MISG images.

To apply the image-guided search to one of the three MISG images (m/z-image, community image or custom selection image), the respective image must be copied into the cache-lasso panel. In this panel, a connected area of interest can be defined using a lasso selection tool. After the selection has been made, a pairwise matching procedure between the cache-lasso image and the MISG image of every visible vertex is invoked. The matching procedure binarizes both images and subsequently computes the amount of overlapping active pixels, i.e. the number of pixels that have a value of one after binarization and have the same position in both images. The options menu offers two variables to adjust the matching procedure, the minimum considered intensity and the minimum required overlap. The minimum considered intensity defines the threshold for the binarization. This variable is important to flexibly filter out signals of low intensity that are not considered relevant, e.g. noise signals. The minimum required overlap defines the amount of overlapping active pixels necessary to be considered as a match. Each vertex whose image is considered a match is highlighted in the same way as after a manual selection.

Since the value of a pixel in the DR image and the modality image will probably have a different interpretation than in the MISG images, this direct pixel-to-pixel comparison procedure is not applicable. Therefore a lasso selection on these images is implemented as a binary image. This binary image has ones at each spectral pixel position that matches the lasso selection, and zeros otherwise. Consequently, the minimum intensity for the pairwise matching procedure will only apply to the MISG images of the visible vertices. The overlap will be computed as before. In this way, the other modalities can serve as a guide for the selection of spatial areas of interest.

To allow the exploration of the MISG and the corresponding MISG images within the context of lasso selection, the vertex selections within the graph panel remain

enabled. However, all vertex transformations are deactivated as long as the lasso is active so that the lasso selection context is not lost. This is also the reason why the lasso application for the MISG images is limited to the cache-lasso panel. The other panels need to remain available to display the MISG images resulting from interactions with the graph.

Another problem is that MSI images can vary vastly in size and spatial resolution. Consequently, very large or very small images can become a problem. Very large images will enlarge the image display, leaving only little space for the graph display. Very small images can be problematic to analyze because some patterns may not be perceptible anymore. To address both problems the size of the image display, including all image panels, can be adjusted in the options menu. In addition, each image can be enlarged within a popup modal using the whole size of the browser window. This allows an even more detailed examination of individual images and also supports a more precise application of the lasso selection tool.

The list display

The list display is closely linked to the graph and the image display and consists of two different elements:

1. The m/z list (see Figure 4.17b),
2. The quantitative graph properties (QGP) list (see Figure 4.17a).

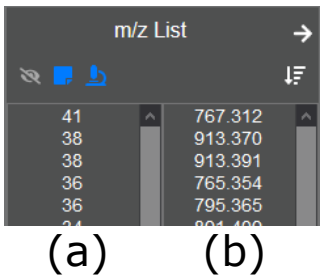


Fig. 4.17.: Snippet to illustrate the structure of the list display – (a) QGP list (only active after the execution of a QGP query). (b) m/z-list (always active, but can be collapsed on demand).

By default, the m/z-list shows all m/z-values provided by the MISG. It is visible by default but can be collapsed if not needed.
The connection between the m/z-list and the graph display can be described as a link, brush and filter connection. To supplement the graph panel with additional information, a selection of vertices filters the m/z-list to show only the m/z-values

that are associated with the selection. The other way around the m/z-list can also be used as a filter. If one or more m/z-values are selected, the corresponding vertices in the graph panel are highlighted using the shape transformation. If an m/z-vertex of one of the selected m/z-values is hidden, the corresponding community node is transformed. Thus the m/z-list not only adds important information to the graph panel, which would otherwise overload it but also allows a fast and targeted search for specific m/z-vertices.
The connection between the m/z-list and the image display is designed to be coherent to the connection between the graph display and the image display. This means that if a single m/z-value is selected, the respective m/z-image and the corresponding community image are shown, while for multiple selected m/z-values the respective aggregated image is shown.
In order to integrate expert knowledge, the m/z-list offers an annotation functionality. Thus, each m/z-value can be supplemented by manual annotation. These annotations are also saved when the export functionality is used.

Quantitative Graph Properties (QGP) The term quantitative graph property (QGP) refers to the output of any function that can be applied to quantify a structural property of a graph. This includes measures such as degree, clustering coefficient, path lengths and many others [102]. Depending on the function, the quantified properties can relate to individual vertices or edges, groups of vertices or edges, diverse combinations or the entire graph. For the QGP-based edge reduction methods (ER-ACC, ER-SUM and ER-PCA), only functions that quantify the properties of the entire graph were used. In contrast, GRINE only implements functions that quantify the properties of individual vertices. When applied to a graph-based network, the interpretation of the different QGP results may depend on the context of the network. For example, the shortest distance between two vertices in a social network may be interpreted as an indicator of how close two people are to each other, while the same property in a transport network may be interpreted as an indicator of the travel distance between two stations.

The QGP list is only activated if a QGP query is executed. The options menu offers a selection of different QGP queries, which will compute the selected QGP function for all m/z-vertices. The results are presented in descending order on the QGP list. To keep the coherence the m/z-list is also sorted accordingly. To allow quick access to both low and high results, the sorting can also be reversed. The behavior of any link, brush, and filter operation is the same as for the m/z-list.

The utilization of QGP queries on MISGs is still very experimental. The basic idea is to support the analysis of MISGs by suggesting vertices with special characteristics.

This includes tasks such as the improvement of community structures, the detection of special graph structures (see Figure 4.18), the identification of biologically more relevant vertices and the assignment of more biological meaning to the vertices in the graph. In addition, the use of QGPs is intended to enable more targeted analysis by suggesting interesting vertices, where the meaning of interesting depends on the particular QGP. Currently, there are not yet enough experiments with MISGs to draw final conclusions about the interpretation of individual QGPs in this context.

There are nine different QGPs currently integrated into GRINE. Possible interpretation approaches, based on initial experiments and theoretical considerations, are outlined in the following.

Degree A high degree of an m/z-vertex may indicate that the molecular distribution of the associated m/z-image is similar to many other molecular distributions. This leads to two possible conclusions, either the particular distribution occurs frequently throughout the data set, or the particular distribution represents a very generic pattern that is sufficiently similar (larger than the similarity threshold γ) to a variety of other distributions.

Average edge weight The average edge weight indicates the strength with which an m/z-vertex is bound to its neighborhood. If this value is high and the degree is not too low, this measure may indicate that the associated molecular distribution is a characteristic distribution for the data set. However, regardless of the degree, this measure still refers to m/z-vertices that show strongly connected groups.

Average neighbor degree A low average neighbor degree may indicate that the m/z-vertex is part of a loosely connected subgroup or chaining structure, especially if this is associated with a low cluster coefficient. Otherwise, a low value may also refer to a small subset of m/z-vertices. If a small value is associated with a high cluster coefficient, it may be an indication of a strongly connected sub-community that can refer to a small set of m/z-images with a distinct molecular distribution. If the value is high, this can indicate either an integration of the m/z-vertex in a large strongly connected subgroup or an offshoot structure. More information can be obtained by considering a high value in combination with the degree of the m/z-vertex.

Cluster coefficient [129] A high cluster coefficient indicates that the m/z-vertex is part of a densely connected subgroup, i.e. a set of m/z-images that are very similar in their molecular distribution. We refer to these densely connected subgroups as cores. Cores can lead to the identification of sub-communities, i.e. a set of m/z-

images within a community which are more similar to each other than to the other m/z-images within the community.

Within group degree A high within group degree, where a group is defined by a community, may indicate m/z-vertices associated with m/z-images that represent molecular distributions that are very characteristic for their community. Therefore, the m/z-images of these m/z-vertices may refer to specific morphological regions in the sample or to molecules important for any molecular processes.

Between group degree A high between group degree, where a group is defined by a community, may indicate that the m/z-vertex and the associated molecular distribution are part of the wrong community. However, if the between group degree is similar to the within group degree, the m/z-vertex may be associated with a molecular distribution that represents some kind of transitional distribution. Molecules with such transitional distributions could refer to transitional states in some molecular process.

Group degree The group degree is defined as the ratio between the squared degree of the m/z-vertex and the total number of edges of the community. A small group degree means that the number of edges of the m/z-vertex is small compared to the total number of edges within the community. This can lead to m/z-vertices that are only loosely integrated in their community, which can indicate offshoot structures or m/z-vertices that are part of the outer sphere of a community. Offshoot structures usually indicate that the m/z-vertex is better removed from the community to improve the overall quality of the community image. m/z-vertices of the outer sphere may indicate molecular distributions that are not as characteristic for the data set as others.

Centrality [33] The centrality is defined as the sum of the fraction of all shortest paths between any pair of vertices passing through the considered vertex. Assume that the path from one m/z-vertex to another can be interpreted as the transition of one molecular distribution to another, i.e. a process of spatial transformation. Then, a high centrality indicates that many spatial transformations between two molecular distributions must pass through the molecular distribution associated with the m/z-vertex of high centrality. Such a molecular distribution may have a central position within the overall molecular flow.

Within cluster centrality The within cluster centrality is defined as the sum of the fraction of all shortest paths between any pair of vertices within the same community that pass through the considered vertex, which is also part of the same community, divided by the degree of the considered vertex. If the within cluster centrality is

high, this indicates that the m/z-vertex is central within its community but has few edges. Such m/z-vertices may lead to bridge structures, i.e. molecular distributions connecting two sub-communities. Such vertices can indicate positions within a community where a split might be appropriated to improve the overall quality of the community detection result.

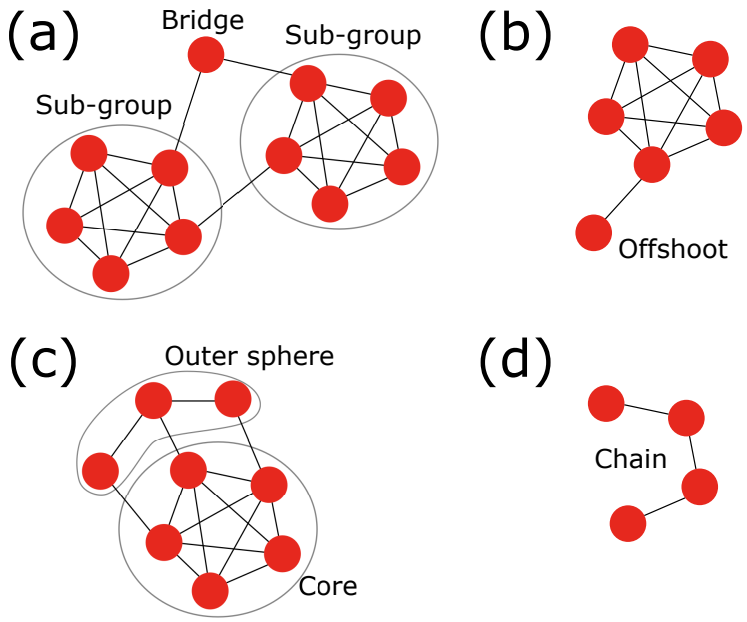


Fig. 4.18.: Examples of special graph structures – (a) A bridge structure. The bridge vertex connects two sub-groups of vertices into one community. (b) An offshoot structure. The offshoot vertex is only loosely connected to the community and can decrease the overall quality of the community. (c) An outer sphere area. The community shows a densely connected core and a loosely connected outer sphere area. (D) A chain structure. The vertices are assigned to the same community but they are not well connected.

The above discussion of QGPs is only is based on first results of our research and the interpretations are rather imprecise. However, it shows that there is potential in the use of QGPs for the analysis and interpretation of MISGs and thus for MSI data in general. However, this topic requires much more research.

Advantages of the graph representation There are many different clustering approaches that can be used to compute clusters of similar m/z-images without the need for any data mapping or transformation. Some of them were already mentioned in Section 4.2.7. However, the transformation into a MISG provides some advantages. Due to the connecting edges, the relations between the individual m/z-images, as well as the results of the community detection can be easily traced. This can offer

advantages during analysis and for interpretation. Likewise, we suspect that the use of QGP queries has the potential to support analysis and interpretation.

Application on Real Data

This section presents an application of community detection on the barley seed data set ($\mathcal{I}^{\mathrm{B}}$). A combination of the Pearson correlation coefficient and ER-PCA was used to create the MISG and the *Louvain method* was applied for community detection. The evaluation uses the identified metabolite classes for the m/z-values of the data set, taken from the work of Gorzolka et al., 2016. At the beginning of this chapter, we made the assumption that metabolic processes are highly spatially localized. For a simply structured organism like the barley seed, it is reasonable to assume that some of the different metabolic processes are spatially separated from each other. Since different metabolic processes require and process different classes of metabolites, it can be assumed that metabolites of the same class are similarly distributed. A good community detection result should reflect this assumption.

The result of the community detection shows an organization in two hierarchies, as illustrated in Figure 4.19. Table 4.4 presents the composition of the metabolite classes per community. It can be seen that most communities are dominated by one class of metabolites. Furthermore, the two carbohydrate communities ($C_{\{1,8\}}$ and C_{10}) and the two hordatine communities (C_4 and $C_{\{5,11\}}$) of the highest hierarchical level are directly connected, which is an indicator for inter-community similarity. Figure 4.20 illustrates how the graph representation can support a detailed analysis of community detection results. The graph structure, shown in Figure 4.20a reveals that the only known carbohydrate in community C_1 is directly connected to community C_8, which consists only of carbohydrates. As this is the only m/z-vertex with an edge to the carbohydrate community, this may be an indication that the m/z-values with unknown classes in C_8 are either not carbohydrate or carbohydrate with a slightly different spatial distribution. The graph structure shown in Figure 4.20 b shows that the only non-lipid in community C_4 is visibly distanced from the other vertices in the community. This shows that even if the grouping by molecular classes is not perfect, the graph structure can help to identify potential uncertainties.

In summary, it can be said that the application of community detection on the barley seed data set yields a qualitatively good result under the given assumption of spatial localization.

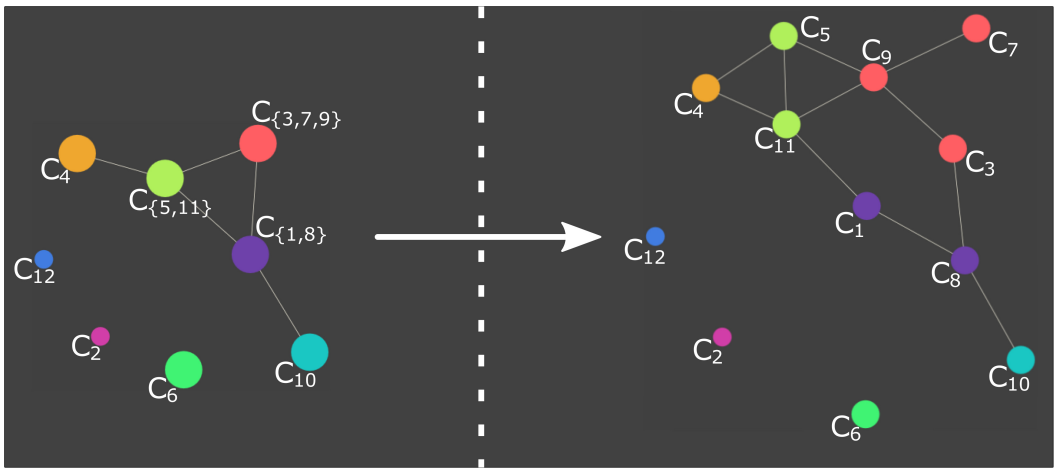


Fig. 4.19.: Result of the hierarchical community detection on the barley data set $\mathcal{I}^{\mathrm{B}}$ – The computation used the Pearson correlation coefficient as the similarity function, ER-PCA for edge reduction and the *Louvain method* for community detection. The left side shows the highest hierarchical level. The right side shows the level below.

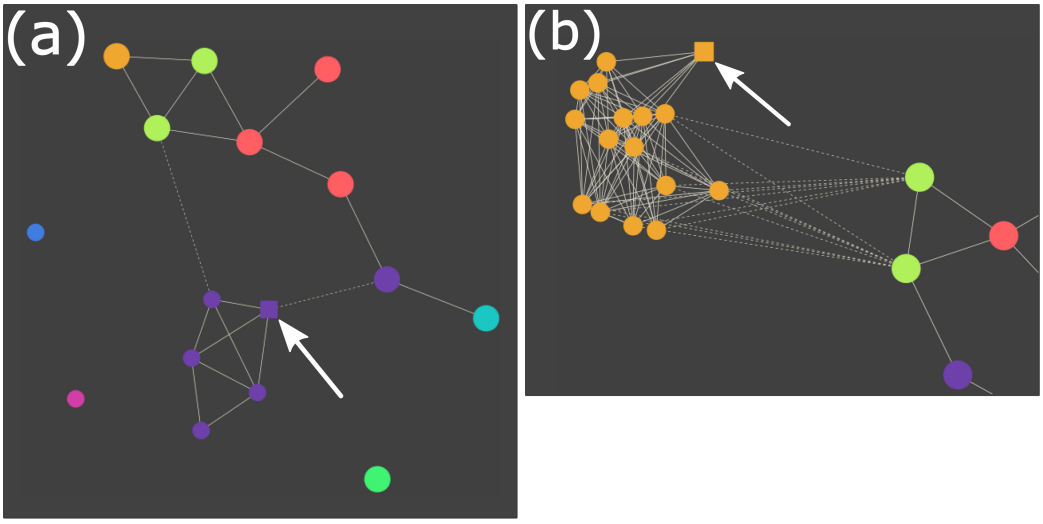


Fig. 4.20.: Illustration how the graph representation can support a detailed analysis of community detection results – The community numbering is the same as in Table 4.4. (a) The only known carbohydrate in community one is directly connected to community eight, which consists only of carbohydrates. (b) The only non-lipid in community four is noticeably distant from the other vertices in its community.

Tab. 4.4.: Summary of the total number of metabolite classes per community – The community detection resulted in two hierarchical levels. The communities are grouped according to this hierarchical organization (see Figure 4.19). The abbreviations for the molecular classes are: carbohydrates (C), lipids (L), hordatines (H) and unknown (U).

Community	# C	# L	# H	# U
Community 1	1	1	0	3
Community 8	10	0	0	0
Sum	11	1	0	3
Community 2	0	0	0	1
Community 3	0	7	0	0
Community 7	0	1	0	1
Community 9	0	3	0	3
Sum	0	11	0	4
Community 4	1	0	15	0
Community 5	0	0	13	0
Community 11	0	1	26	0
Sum	0	39	0	0
Community 6	0	3	0	0
Community 10	10	0	0	0
Community 12	0	0	0	1

4.3.3 Improvements and Future Research

The most essential step to project a set of m/z-images onto a graph structure to create a MISG is the edge reduction step. But the definition of a threshold, that decides which edges should be removed, is anything but trivial. To address this problem we proposed four different approaches with individual strengths and weaknesses. Nevertheless, we are confident that there is still room for improvement. This includes the definition of the threshold value, but also the investigation of a possible dependence of such a threshold value on the data set, similarity function and community detection method.

In our experiments, we investigated a handful of community detection methods, of which the most effective one was presented in this chapter. However, the detection of communities is a constantly evolving topic. A more extensive study to examine the differences between the various community detection approaches could lead

to improved results. Another interesting approach could be to use a variety of community detection methods to build consensus communities to compensate for individual methodological weaknesses. A similar idea has already been used to compensate for the different results with different random seeds in the context of community detection, which led to good results [58, 116]. The idea could also be extended to compute consensus communities not only for different community detection algorithms but also based on different similarity functions.

There are also improvements to GRINE that could increase its analytical effectiveness. This includes the implementation of the ability to create more than one NodeTrix representation simultaneously. In addition, the order of vertices in the NodeTrix representation is currently based on community membership, followed by the m/z-value. An order based on similarity, similar to the order of a dendrogram of agglomerative hierarchical clustering, should improve the analytical effectiveness and efficiency of the NodeTrix representation. This is especially true concerning the identification of sub-communities.

As mentioned above, we think that there is potential in the use of QGPs to support the structural analysis and biological interpretation of MISGs. However, the interpretation of the different QGPs in the context of MISGs is still at an early stage and requires further research.

Another future research topic could investigate how GRINE can be extended to enable the visual analysis of overlapping community structures. For example, by including multi-colored vertices. Furthermore, there is an open research question regarding the comparison of MISGs. The visual comparison of multiple graphs is a difficult problem. A first approach could investigate the use of multiple visual layers, i.e. one for each graph, combined with link, brush and filter operations. For an algorithmic comparison, we have already examined several approaches, such as graph edit distance, spectral distance, community overlaps, and more, but none of these approaches led to satisfying results. Therefore, further research is needed to investigate the potential value of the algorithmic comparison.

4.4 Approximation of Regions of Interest in the Spatial Domain

A commonly applied method in MSI that can be used to propose regions of interest after clustering of mass spectra is the computation of segmentation maps [4, 7, 5, 36, 12, 26] (details about segmentation maps are discussed in Section 5.3). However, this method is not applicable to the spatial domain, since no individual pixels are clustered. Therefore, we propose a new method to approximate regions of interest for clusters of m/z-images.

The complete method is presented in Figure 4.21 and can be summarized as follows:

1. Compute a representative image for each cluster.
2. Approximate regions of interest using the maximally stable extremal regions (MSER) method [70] on the representative images.
3. Refine the approximated regions using an active contour algorithm (MorphACWE) [69].

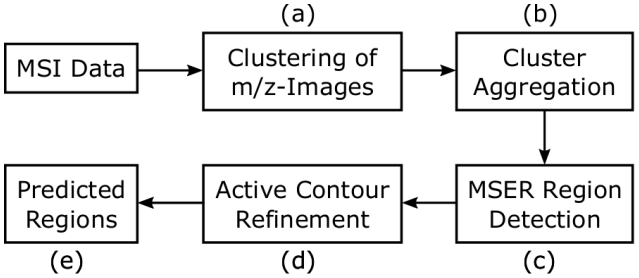


Fig. 4.21.: Outline of the spatial region prediction method – (a) m/z-images are clustered by an arbitrary clustering method. (b) A representative image for each cluster is computed by an aggregation method, such as the arithmetic mean. (c) The maximally stable extremal regions (MSER) method is used to approximate interesting regions. (d) The approximated regions are refined, using an active contour method. (e) The predicted interesting regions are returned.

Remark: All m/z-images for this application are normalized to 8-bit, i.e. $\mathcal{I}_{i,j,z} \in \mathbb{R} \mapsto \mathcal{I}_{i,j,z} \in \{0, 255\}$ using Equation (4.3) with $a = 0$ and $b = 255$.

For the computation of cluster representatives, the proposed method uses the arithmetic mean of all m/z-images in a cluster.
For the initial region prediction, an in-house implementation of the MSER method is used. The idea of the MSER method as applied in the in-house implementation is to detect sequences of extremal continuous (i.e. connected) regions $\mathfrak{S}_i =$

$\{\mathfrak{s}_j, \ldots, \mathfrak{s}_{q-\mathfrak{d}}, \mathfrak{s}_q, \mathfrak{s}_{q+\mathfrak{d}}, \ldots, \mathfrak{s}_{j+|\mathfrak{S}_i|}\}$, with $j, q \in \{0, 255\} \wedge j < q$ in continuously thresholded binary images using a series of thresholds $q \in \{0, \ldots, q + \mathfrak{d}, q + 2\mathfrak{d}, \ldots, 255\}$, with $\mathfrak{d} \in \{1, 254\}$. Whereby an extremal continuous region is defined as a region of active (1) binary pixels that cannot be extended at their outer edge by a von Neumann neighborhood adjacency relation. Within a sequence $\mathfrak{S}_i$, a region is maximally stable if the relation $|\mathfrak{s}_{q-\mathfrak{d}} \setminus \mathfrak{s}_{q+\mathfrak{d}}| \,/\, |\mathfrak{s}_q|$ has a local minimum at q.

Thus, the in-house implementation of the MSER method depends on four parameters: the minimum required length of the sequence of nested extremal regions $|\mathfrak{S}_i|$, the value increase (intensity value/color value) for each iteration $\mathfrak{d}$, the minimum region size $\mathfrak{a}^{\text{min}}$ and the maximum region size $\mathfrak{a}^{\text{max}}$, where $\mathfrak{a}^{\text{min}}$ and $\mathfrak{a}^{\text{max}}$ can be used to remove regions which are too small or too large.
The method is very sensitive to all of these parameters and requires careful tuning.

However, independent of the tuning the regions computed with the MSER method will often be quite perforated. This might be due to the high pixel-to-pixel variability, which is inherent to many MSI instruments. Therefore, the proposed method uses an active contour method to refine the computed regions. In previous work, active contours showed good results in the refinement of regions of interest in MSI [131]. The proposed method uses the morphological active contours without edges (MorphACWE) method of scikit-image [127].

Remark: The clustering was applied using only the spectral pixel ρ of the m/z-images. However, the MSER and the MorphACWE method were applied to the whole set of pixels p.

An example of the proposed method is shown in Figure 4.22. It shows four example clusters of an agglomerative hierarchical clustering result on the barley seed data set $\mathcal{I}^{\text{B}}$. The clustering method was applied using the correlation distance, average linkage and an approximated number of 14 clusters according to Equation (4.27). The MSER method was applied to the arithmetic mean image of each cluster with $\mathfrak{s} = 10, \mathfrak{d} = 1, \mathfrak{a}^{\text{min}} = 0.03$ (173 pixel) and $\mathfrak{a}^{\text{max}} = 0.5$ (2886 pixel). The MorphACWE was applied with 1000 iterations and the number of smoothing operator applications per iteration was set to two. The detected regions of the MSER method were used as the initial level sets.

Although the m/z-images are quite noisy, it can be seen that the region prediction is capable to approximate distributions that have intense signal strengths in most m/z-images of the cluster. It can also be seen that these regions can be disjunct (cluster zero) but also nested (cluster thirteen). However, especially for noisy images, the method might return regions that are based on noise, such as the lower

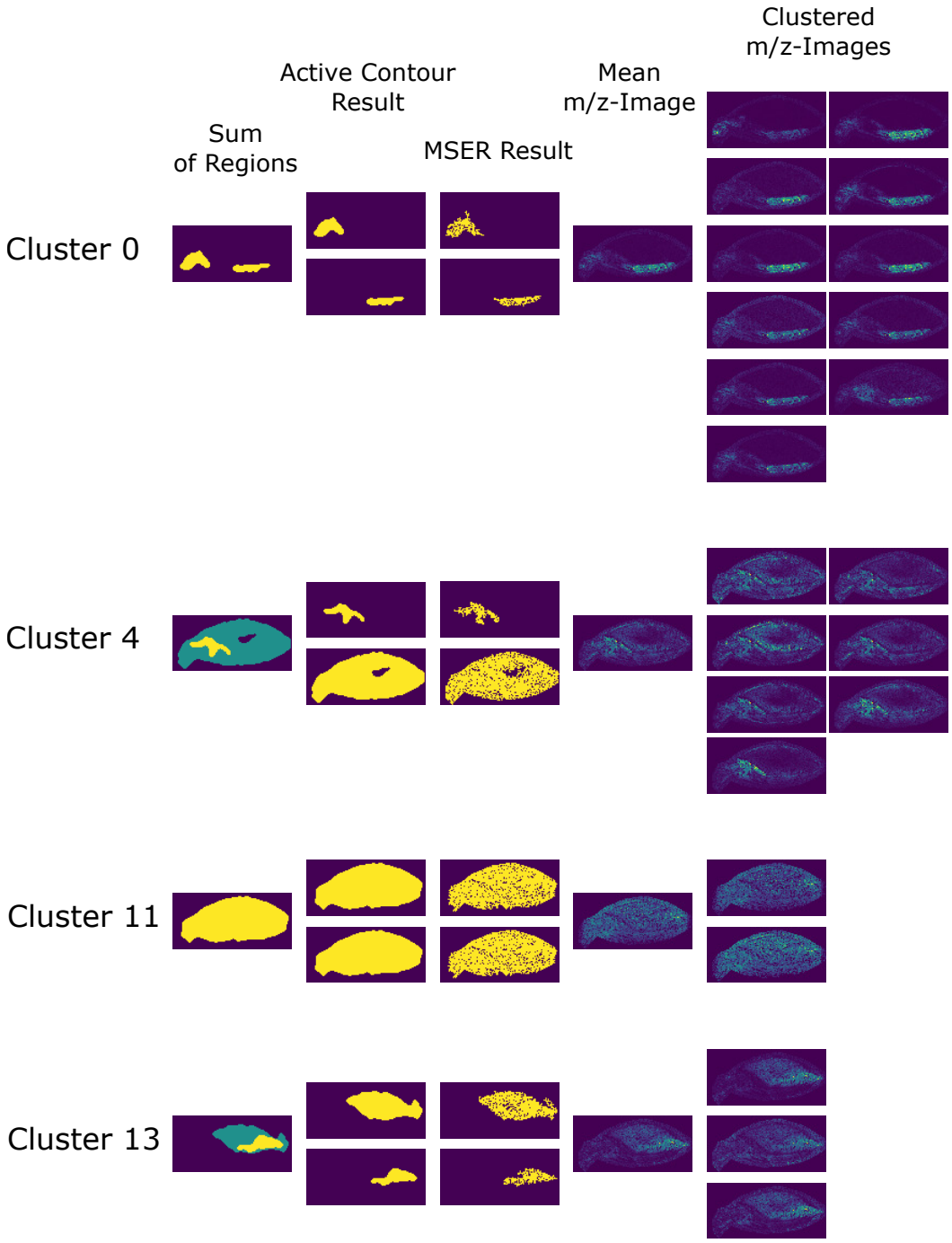


Fig. 4.22.: Example application of the spatial region prediction – Predicted regions of interest for four m/z-image clusters of the barley seed data set $\mathcal{I}^{\mathrm{B}}$. The first column shows a region overview (sum of regions). The second column shows the result of the MorphACWE active contour method. The third column shows the result of the MSER method with $\mathfrak{s} = 10, \mathfrak{d} = 1, \mathfrak{a}^{\min} = 0.03(173\text{pixel}), \mathfrak{a}^{\max} = 0.5(2886\text{pixel})$. The fourth column shows the arithmetic mean image of the respective cluster. The fifth column shows the individual m/z-images of the respective cluster. The applied clustering method was agglomerative hierarchical clustering, using correlation distance, average linkage and a predicted number of 14 clusters according to Equation (4.27).

approximation in cluster four or the approximations in cluster eleven. A proper pre-processing of the m/z-images might solve this problem. Another weakness of the proposed method is that duplicates of regions can emerge (cluster eleven), as well as nested regions with minor differences. An additional filter step after the detection of maximally stable regions might be appropriate to compensate for this weakness. A further weakness is that a proper selection of $\mathfrak{a}^{\max}$ can be unintuitive, due to the situation that ρ can be much smaller than p, which is given for the presented example. Furthermore, some experiments with further data sets (not shown) returned approximated regions that seemed to be incomplete or not associated with any morphological structure of the sample. However, this could be a consequence of noise, missing or inappropriate pre-processing and the approximated regions are of course strongly dependent on the result of the clustering and the computation method used for the cluster representatives.

In summary, the presented method seems to be a promising approach to approximate regions of interest for clusters of m/z-images. However, the method is still experimental and has some major weaknesses that need to be addressed. Thus, more improvements and research are required to make a final statement about the effectiveness of the method.

4.5 Summary and Contributions

In the first part of this chapter, we introduced a workflow scheme called SoRC to evaluate and optimize different pipeline setups to quantify the similarity between m/z-images for the purpose of clustering. Using SoRC, a variety of pipeline setups have been investigated using different pre-processing techniques, similarity functions and clustering methods. The results demonstrated that pipeline optimization can have a positive impact on cluster results, especially when the spatial irregularity of the data increases. Furthermore, we provided a comprehensive comparative analysis of different similarity functions to quantify the similarity between m/z-images considering different degrees of regularity.

In the second part of this chapter, we proposed to use community detection as a new approach to cluster m/z-images. We presented a method to project the relationship of a set of m/z-images onto a graph structure and used community detection to identify clusters of co-localized molecules. The suitability of this approach was demonstrated using the barley seed data set. We also presented GRINE, an interactive visual analysis tool for the exploration and analysis of communities in m/z-image similarity graphs. The use of a docker container makes GRINE easy to use and allows it to run independently from the operating system.

In the third part of this chapter, we proposed a method to approximate regions of interest based on clusters of of co-localized m/z-images.

5 Comparative Molecular Composition Analysis in the Spectral Domain

Code: `https://github.com/Kawue/whide-v2/`

> *Science has made us gods even before we are worthy of being men.*
>
> **— Jean Rostand**
> (Biologist and philosopher)

5.1 Motivation

The comparative analysis of molecular composition focuses on the comparison of different mass spectra. A mass spectrum represents the molecular composition of a measured analyte. In MSI, each measured position on the biological sample can be considered a single analyte. Since each measured mass spectrum is assigned to a defined location (pixel), this means that each mass spectrum represents the location-dependent molecular composition within a sample.

The comparative analysis of molecular compositions is frequently applied for supervised and unsupervised analysis of MSI data. The two primary objectives are classification, i.e. the assignment of spectra (pixel) into predefined classes (supervised analysis) and the segmentation, i.e. the assignment of spectra (pixel) into newly defined groups (unsupervised analysis) [73, 25, 4, 7, 54, 12].

This chapter focuses on the unsupervised clustering of mass spectra (pixels) for the computation of segmentation maps and the visual analysis of these segmentation maps. For this purpose, two clustering approaches to create segmentation maps are discussed. Then, a visual analysis tool for exploring and analyzing segmentation maps is presented and the added value achieved through interactivity is demonstrated and discussed.

5.2 Hierarchical Hyperbolic Self Organizing Maps

A self-organizing map (SOM) is a type of artificial neural network and was originally proposed by Kohonen in 1982. The SOM was proposed for dimension reduction and clustering [53]. It is trained by unsupervised competitive learning to generate a descriptive model of the input data in a low-dimensional space while preserving topological properties. This descriptive model is also called "map". A map consists of several so-called neurons or neural units, which are organized in a grid. A map can be formally described by a set of neurons $\Psi = \{(u_r, b_r)_{r=1,\dots,R}\}$, where b_r denotes the grid position and u_r is a weighting vector, also called neural unit. Each neural unit can be considered as a prototype, which is a representative of a set of data points (feature vectors). The grid structure defines the position b_r of each neuron on the map and the weighting vector u_r describes the position of each neuron in the input space. The exact grid structure differs for the various types of SOMs.
In order to use a SOM for clustering, it has to be trained. This means that the weighting vectors of the neurons are adjusted to fit their position to the input space. The objective of the training is to resemble the topology of the input space. This is realized by minimizing a predefined distance function δ. After the grid structure has been created, the training can be described as follows:

1. Select a single feature vector x_q of the data set $\mathfrak{X}$ (e.g. randomly).

2. Determine the neural unit with the minimum distance to the current feature vector x_q according to the predefined distance metric. This neural unit is referred to as the best matching unit (BMU).

3. Update the BMU according to: $u_r^{(a+1)} = u_r^{(a)} + \varepsilon^{(a)} \cdot \tau^{(a)}(q, r) \cdot (x_q - u_r^{(a)})$, where $\varepsilon^{(a)}$ is the learning rate and $\tau^{(a)}(q, r)$ is a neighborhood function for training iteration a.

4. The neighbors of the BMU are also adapted but to a lesser degree, which is controlled by the neighborhood function.

5. Repeat for A iterations or until a stopping criterion is reached. Usually, the adaption rate, i.e. the learning rate and the range of the neighborhood function, decreases throughout the training process to prevent overfitting.

For the purpose of clustering the training is performed with every feature vector $x_q \in \mathfrak{X}$. After training, each neural unit u_r represents the prototype of a unique cluster. Subsequently, each feature vector $x_q \in \mathfrak{X}$ is mapped to its corresponding BMU.

The hierarchical hyperbolic self-organizing maps (H^2SOM) is a development of the self-organizing map (SOM) [84]. In contrast to the SOM, the topological structure of the H^2SOM is not embedded in Euclidean space, but in hyperbolic space. Furthermore, the neurons are arranged in several circular and hierarchically organized layers, called rings $\mathfrak{r}_i$ with $i \in \{1, \ldots, \mathfrak{R}\}$. The rings are arranged around a central neuron ($\mathfrak{r}_0$). Each neuron has exactly $\mathfrak{n}$ neighbors, except for the outermost ring, where each neuron has exactly three neighbors. Two neurons are defined as neighbors if they share a connection due to the grid structure. The neighborhood of every neuron of an inner ring ($\mathfrak{r}_{1,\ldots,\mathfrak{R}-1}$) consists of one neighbor in the last ring (parent), two neighbors in the same ring (siblings) and $\mathfrak{n} - 3$ neighbors in the next ring (children). The grid of the H^2SOM is built using the Möbius transform. The construction starts at the central neuron, around which $\mathfrak{n}$ children are spawned. From there the neurons are traversed recursively. At each neuron $\mathfrak{n} - 3$ children are spawned until every neuron has $\mathfrak{n}$ neighbors (except for the outermost ring) and the maximum number of $\mathfrak{R}$ rings is reached. This procedure is illustrated in Figure 5.1. More detailed information can be found Ontrup and Ritter, 2006. The training for the H^2SOM is performed ring-wise, i.e. all the training steps described above are performed for each ring individually. The neighborhood function for the training of the H^2SOM is defined as follows:

$$\tau^{(a)}(q,r) = \exp\left(-\frac{\left(2\,\mathrm{arctanh}\left(\left|\frac{b_q - b_r}{1-\bar{b}_j b_r}\right|\right)\right)^2}{\mathfrak{w}^{(a)^2}}\right) \tag{5.1}$$

where $\mathfrak{w}^{(a)}$ is the width of the neighborhood bell function in iteration a.

The number of neurons on the grid grows exponentially with an increasing number of rings. This can lead to a more trustworthy reconstruction of the input space topology by the grid structure and thus to a more trustworthy representation of the feature vectors by the BMUs (training and mapping). However, it can also lead to an overestimation of the intrinsic dimensionality of the input space, thus reducing cluster and mapping performance.

The increased number of neurons also dramatically increases the computation time required for searching the BMUs during training. To address this problem, a beam search can be applied to approximate the BMU at each training step, exploiting the hierarchical structure of H^2SOM. The beam search starts with the central neural unit as the initial BMU. For a beamwidth of $\mathfrak{b} = 1$, the search recursively chooses the next BMU among the children of the last BMU until it reaches the periphery of the current ring to be trained. For $\mathfrak{b} > 1$, the procedure operates equally, but the children of the first $\mathfrak{b}$ BMUs are examined recursively. This strategy has shown to

significantly accelerate the training process while staying close to the performance of the global search [84].

The performance of the H^{2}SOM depends on parameters that need to be optimized. The parameters are the learning rate $\varepsilon^{(a)}$, the width of the neighborhood bell function $\mathfrak{w}$, the total number of rings $\mathfrak{R}$ and the number of neighbors $\mathfrak{n}$. While the algorithm is rather robust against changes in $\varepsilon^{(a)}$ and $\mathfrak{w}$, it is quite sensitive to changes in $\mathfrak{R}$ and $\mathfrak{n}$.

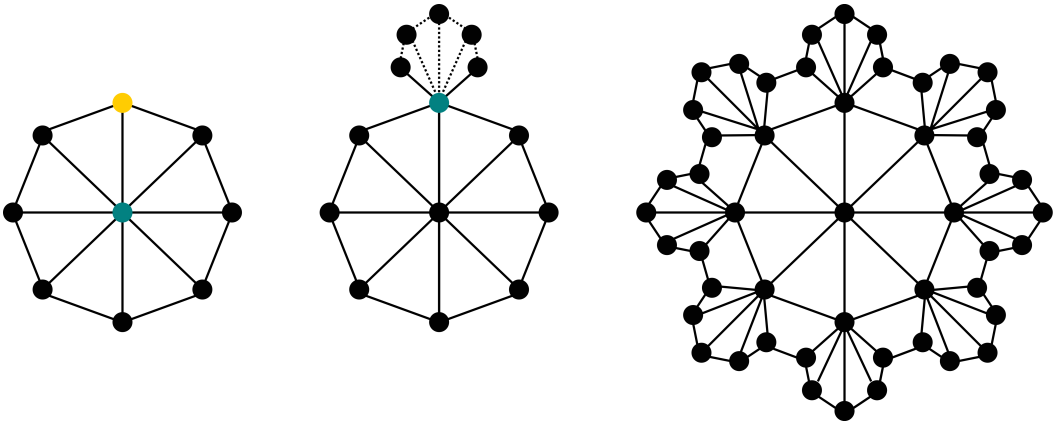


Fig. 5.1.: Example of the H^{2}SOM construction using the Möbius transform – The number of rings and neighbors equal $\mathfrak{R} = 2$ and $\mathfrak{n} = 8$. Teal and yellow color indicate the current and next center neurons. The first grid illustrates the initialization process of the first ring. The second grid illustrates the first construction step of the second ring and the third grid illustrates the final H^{2}SOM.

5.3 Segmentation Maps

A segmentation describes the assignment of a set of pixels into newly defined groups. This process is also called partitioning of an image. The segmentation process assigns a label to each pixel. An assignment to the same label usually reflects shared features. These features can either be derived directly from the image or be the result of further computations. The different labels can then be visualized by unique colors. This type of visualization of a segmentation result leads to a pseudocolored image, which is also known as segmentation map. Segmentation maps are usually used to either simplify the presentation of an image or to represent the relationship between certain regions based on a set of analyzed features. To compute the segmentation of an image, pixel clustering is a commonly used technique. This also includes multivariate images, such as MSI data. Segmentation maps therefore offer an effective method to visualize clustered (multivariate) image data.

In the context of the clustering of MSI spectra (spectral pixel), the fundamental idea is to project the clustering result onto the morphology of the sample, as illustrated in Figure 5.2. This means that if pixels share the same color, it indicates that they share similar spectra (similar molecular compositions), at least according to the clustering criterion. This way, a segmentation map visualizes an abstraction of an MSI data set with just one image. The use of segmentation maps as a visualization technique for clustering results offers several benefits. It supports the identification and definition of unique and possibly unknown regions and also facilitates the interpretation of the clustering result by associating it with knowledge from the biological morphology.

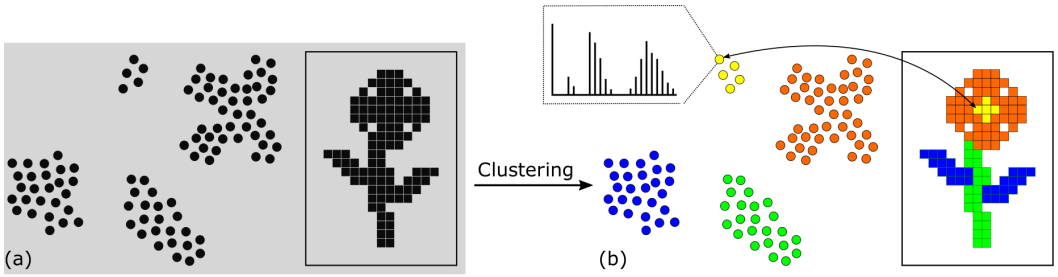


Fig. 5.2.: Illustration of the segmentation map concept – (a) shows the input data. Each data point represents a mass spectrum and is associated with a fixed position on the sample (see frame). (b) shows the grouping into four clusters. Each cluster is assigned a unique color. The corresponding coloring of the positions on the sample (pixels) creates a pseudocolored representation. The result is a projection of the clusters onto the sample.

5.3.1 H^{2}SOM Projection

As discussed above, the H^{2}SOM is embedded in hyperbolic space instead of Euclidean space. This poses a special problem for visualization since the vast majority of display devices (digital and analog) are based on the two-dimensional Euclidean space. However, there exists no truthful representation of $\mathbb{H}^2$ in $\mathbb{R}^2$, since the projection of the negatively curved hyperbolic space into the flat Euclidean space introduces distortions in either length, area or angle [84]. However, the Pointcaré disc model provides a projection of the $\mathbb{H}^2$ into the unit disc. This projection has several benefits:

1. It provides a representation of the $\mathbb{H}^2$ in $\mathbb{R}^2$, which allows an intuitive visualization on a two-dimensional display.

2. The projection is locally shape-preserving, exhibiting a strong “fish-eye” effect. This means that the origin of the $\mathbb{H}^2$ is presented almost faithfully, but distant regions become exponentially squeezed.

3. Möbius transformations can be applied to translate the original $\mathbb{H}^2$ space. This allows to access details at the squeezed borders by changing the focus, i.e. the center is moved to selected areas of the $\mathbb{H}^2$.

The presentation onto a circle provides a convenient benefit for the visual presentation. The hierarchical ring-wise grid structure can be mapped onto the hue disc of the Hue-Saturation-Lightness (HSL) color space, which is illustrated in Figure 5.3. This way, a distinct color value is assigned to each neuron, which creates colorful segmentation maps. Due to the H^{2}SOM grid structure, similar neurons also correspond to similar colors. However, besides these beneficial properties, the projection onto the hue disc also exhibits a pitfall.

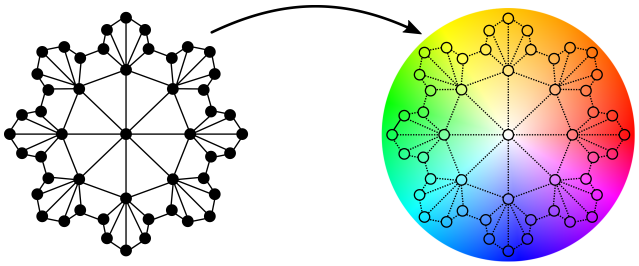


Fig. 5.3.: H^{2}SOM grid projection – Illustration of the projection of the H^{2}SOM grid structure onto the hue disc of the Hue-Saturation-Lightness (HSL) color space.

The perceptual changes in the HSL color space are not uniform. This means that equal distances between any two points in the color space are not always perceived to be equal. Some distances can appear smaller and some larger. Therefore, such a segmentation map can be susceptible to the perception of artificial boundaries (discussed in Section 1.4) and to misjudgment of distances between segments (clusters). In the worst case, this can lead to false interpretations and conclusions.

By exploiting the "fish-eye" effect, it is possible to dynamically direct the focus to a specific subset of neurons within the H^{2}SOM grid. Depending on this focus, some neurons are placed in close proximity, while the other neurons are spread across the rest of the hue disc. As a result, neurons in close proximity are assigned to very similar colors, while extended neurons receive a high color contrast.

5.3.2 Ring-wise Position Optimization of the H^{2}SOM Grid Projection

The use of the HSL hue disc has two notable weaknesses. First, the perceptual changes in the HSL color space are not uniform. This can lead to the perception of artificial boundaries and a misjudgment of the distances between the segmented

areas of the segmentation map. Second, the distance between any two adjacent neurons within each ring is constant. However, the weighting vectors, i.e. the prototype vectors of the individual clusters, do not necessarily share such a constant distance. Accordingly, such a visual representation is also susceptible to misinterpretation.

To address this problem, we have developed a ring-wise position optimization algorithm to adjust the position of each neuron on the hue disc. The algorithm is based on the correlation between the weighting vectors of each neuron to its direct neighbors (siblings). The higher the correlation, the closer the neurons become to each other (see Equations (5.2) and (5.3)). The basic concept of the algorithm is shown in Figure 5.4. The position optimization algorithm iteratively adjusts the position b_r of each neuron. Therefore, in each iteration an individual adjustment factor $\mathfrak{c}$ of each neuron is calculated. The algorithm has two different variants, referred to as "winner-takes-all" and "tug-of-war". For the "winner-takes-all" variant only the sibling with the higher correlation influences the position adjustment, whereas in the "tug-of-war" variant both siblings have an influence. For each neuron with position b_r on the hue disc and an associated weighting vector u_r new positions are calculated by Equation (5.2) for the "winner-takes-all" variant and by Equation (5.3) for the "tug-of-war" variant. The correlation is defined by the Pearson correlation coefficient ϑ. The algorithm works iteratively and adjusts the positions for each neuron in each ring for a predefined number of iterations or until the automatic stopping criterion is reached. The automatic stop criterion is reached if $\vartheta(u_q, u_r) < \frac{c}{1+\|b_q - b_r\|_2}$ for all neurons, where $c > 0$ is a constant for tightening or relaxing the condition (default: $c = 1$). By using the "mitigation" parameter, the position adjustment per iteration can be reduced, i.e. the adjustment is performed in smaller steps.

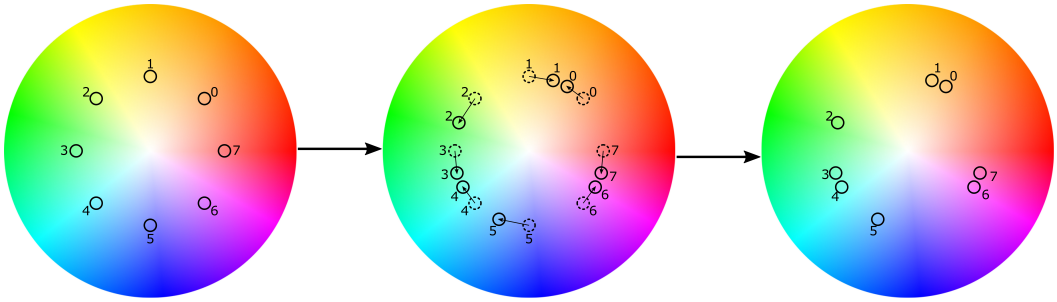


Fig. 5.4.: H²SOM neuron position optimization – Illustration of the position optimization algorithm for the first ring of an H²SOM, with $\mathfrak{n} = 8$ neighbors, after one iteration. The middle illustration shows the change in the position of each neuron. Numbers and arrows are used to facilitate tracking of the relocation. The old and new positions of the neurons are shown as dashed and solid circles.

"winner-takes-all":

$$
\begin{aligned}
q &= \begin{cases} r+1, & \text{if } \vartheta(u_{r+1}, u_r) > \vartheta(u_{r-1}, u_r) \\ r-1, & \text{otherwise} \end{cases} \\
\mathfrak{f}_{r,q} &= \begin{cases} \frac{1}{1+\|b_q - b_r\|_2}, & \text{if the "mitigated" parameter is set to true} \\ 1, & \text{otherwise} \end{cases} \\
\mathfrak{c}_{r,q} &= (b_q - b_r) * \vartheta(u_q, u_r) * \mathfrak{f} \\
b_r &= b_r + \frac{\mathfrak{c}}{2}
\end{aligned}
\tag{5.2}
$$

"tug-of-war":

$$
\begin{aligned}
\mathfrak{f}_{r,q} &= \begin{cases} \frac{1}{1+\|b_q - b_r\|_2}, & \text{if the "mitigated" parameter is set to true} \\ 1, & \text{otherwise} \end{cases} \\
\mathfrak{c}_{r,q} &= (b_q - b_r) * \vartheta(u_q, u_r) * \mathfrak{f}_{r,q} \\
b_r &= b_r + \frac{\mathfrak{c}_{r,r+1}}{2} + \frac{\mathfrak{c}_{r,r-1}}{2}
\end{aligned}
\tag{5.3}
$$

5.3.3 Projection of other Clustering Methods

Retrospectively, the idea of using a circular colormap to visualize a circular, hierarchically organized grid structure of neurons is plausible. Particularly with regard to the additional properties that should be fulfilled, i.e. that clusters of prototypes with high similarity should be colored more similarly and that the color range should be diverse enough to be suitable for a qualitative, categorical coloring, which also does not tempt to a quantitative interpretation. This raised the question of whether this colormapping concept can be transferred to other clustering methods that have similar requirements. To answer this question, we coupled a general clustering approach with an adapted projection method. As a result, the colormapping concept was successfully transferred, which greatly extended the scope of our visual analysis tool (WHIDE; explained below).
The adapted method can be executed in two variants:

1. "cluster-first":
    - Computation of clusters by an arbitrary clustering method.

    - Computation of prototypes, if necessary.
  - Computation of a two-dimensional embedding of the prototypes by an arbitrary dimension reduction method.
  - Projection of the prototypes onto the hue disc (Equation (5.4)).

2. "embed-first":
  - Computation of a two-dimensional embedding of the data by an arbitrary dimension reduction method.
  - Computation of clusters by an arbitrary clustering method.
    - Computation of prototypes, if necessary.
  - Projection of the prototypes onto the hue disc (Equation (5.4)).

The difference between "cluster-first" and "embed-first" is that with "cluster-first" the clustering is computed within the high-dimensional feature space, followed by an embedding of the prototypes into a two-dimensional feature space. For "embed-first" the data is first embedded into a two-dimensional feature space and the clustering is computed on the embedding. Due to their different benefits, both variants have their use case. The "cluster-first" variant has the benefit that the entire information of all provided features can be used by the clustering algorithm. However, as explained in Section 3.1.2 the concept of distance-based neighborhoods becomes more and more meaningless with increasing dimensionality, which is why the "embed-first" variant may be more effective in such cases. The "embed-first" variant may also be more efficient if the computational requirements for the clustering method become too demanding within the high-dimensional feature space.

In general, any arbitrary combination of dimension reduction method and clustering algorithm can be applied with this approach. However, since one of the main considerations for using the hue disc projection is that prototypes with high similarity should be colored more similarly, a dimension reduction method that aims to preserve the topology of the original feature space (e.g. UMAP) should be preferred. For clustering algorithms that are not prototype-based, the prototypes are approximated as the central points within each cluster, i.e. approximated prototypes are the arithmetic mean of all feature vectors of each cluster.

The idea for the actual projection procedure, i.e. the way the prototypes are projected onto the hue disc, was developed in cooperation with Jonas Bicker. The projection can be imagined as drawing a circle around all prototypes. The center of the circle is the arithmetic mean of all prototypes in the two-dimensional embedding. The radius of the circle is given as the Euclidean distance to the furthest prototype including a buffer (see Equation (5.4)). The projection procedure is illustrated in Figure 5.5.

$$\begin{aligned} \overline{b} &= \frac{1}{R} \sum_{r=0}^{R-1} b_r \\ b'_r &= \frac{(1-\mathfrak{a})(b_r - \overline{b})}{\max_r \left\| b_r - \overline{b} \right\|_2} \end{aligned} \tag{5.4}$$

where b_r is a two-dimensional vector specifying the position of a prototype, b'_r is the projected position and $\mathfrak{a}$ specifies the buffer to the edge of the unit circle (default: 0.1).

Compared to the H^{2}SOM projection, this method has an additional weakness that needs to be addressed. The H^{2}SOM grid structure has the convenient property that the neurons are organized in rings, which means that each neuron of the same ring has the same distance from the center. This property is no longer given for the projection of embedded prototypes, which leads to the following problem: Imagine three prototypes with the same distance to each other. If they are mapped in such a way that two of them share the same angle on the hue disc but have different distances to the center, they will be assigned to the same color. This is because the hue disc does encode the angle (by color) but not the distance to the center. This problem cannot be solved by changing the positioning of these prototype without violating at least one of the original relationships. Consequently, this problem cannot be completely solved with the presented projection method. However, luminance can be used to encode the distance from the center. This has the effect that the colors will be different for varying distances from the center. However, a change in hue is likely to be perceived with higher contrast by users than a change in luminance. A change in luminance can also be misinterpreted as a quantitative rather than a qualitative differentiation feature. However, encoding the distance from the center with varying luminance can at least partially compensate for the problem. Consequently, the projection step of the above method descriptions for "cluster-first" and "embed-first" have to be adjusted from hue disc to hue-luminance disc. Although the adapted projection method uses a hue-luminance disc instead of a hue disc, the rest of this chapter will not differentiate between both. To summarize both, from now on only the term color disc is used.

The adapted projection method has two major weaknesses that should be considered. First, if two prototypes are close together in the embedded space, they might be assigned opposite colors. This happens when the center of the color disc is located between them. This weakness cannot be easily compensated with the presented projection method for static visualizations, but it can be addressed for interactive visualizations by relocating the center. Second, depending on the topology of the

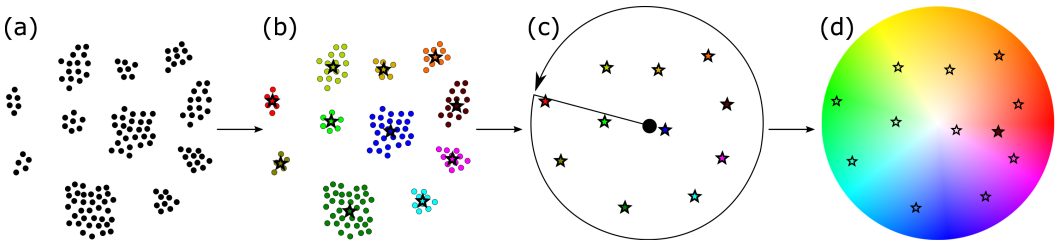


Fig. 5.5.: Illustration of the embedded prototype projection procedure – The illustration shows the "embed-first" approach. For "cluster-first" the steps (a) and (b) cannot be visualized. The procedure for steps (c) and (d) is the same. (a) Two-dimensional embedding of a data set. (b) Result of a clustering, where each color represents a cluster. Prototypes are visualized as stars. (c) Prototype projection onto the unit circle. The center is indicated by a large black dot. (d) Illustration of the projected prototypes on the HSL hue-luminance disc.

embedding, the projection may populate only parts of the full color disc. This means that the colormapping may use less of the color space compared to the H^2SOM projection, which always populates the full color disc.

Remark The application of UMAP becomes computationally time-consuming as the feature space and data points increase. To address this problem, it is possible to use a dimension reduction method with lower computational resource requirements (e.g. PCA) to pre-reduce the feature space (e.g. $\mathbb{R}^q \rightarrow \mathbb{R}^{1000}$).
Consequently, the adapted projection method would change as follows:

- Computation of a pre-reduced feature space using a dimension reduction method with low computational resource requirements.
- Continue with "cluster-first" or "embed-first" using the pre-reduced embedding.

Projection of Dimension Reduction Embeddings

Interestingly, the adapted method for projecting embedded prototypes is also well applicable to project a complete two-dimensional embedding, i.e. all embedded data points. Again, dimension reduction methods that aim to preserve or reproduce the topology of the original high-dimensional feature space are better suited for this type of visualization. The visualization generated with this approach has to be interpreted differently than the previous segmentation maps. The result is no longer a segmentation map in the actual sense since each pixel is treated as a separate cluster. However, it can be interpreted as a kind of soft segmentation map. The closer two pixels are in the embedded space, the more similar their color will be. The same applies to the opposite, the farther two pixels are apart in the embedded space, the more different their color will be. If the topology of the original high-dimensional

feature space is well preserved, pixels with similar molecular composition are likely to be close together. In such cases, the visualization will create multiple regions, each consisting of very similarly colored pixels, although most pixels will be assigned to a unique color. It is also likely that the differently colored regions are connected by intermediate regions which will cause a gradual change in color.

Remark After the projection method has been applied, the distance from a data point to the center of the color disc is bounded by the interval $[0, 1]$. To avoid colors that are too close to pure white or pure black, these boundaries can be mapped to a desired smaller interval, e.g. $[0, 1] \mapsto [0.25, 0.75]$, which will be the case for the results presented in Section 5.3.4.

5.3.4 Application on Real Data

Remark In the following, different segmentation maps are compared. For the comparison of different segmentation maps, it has to be considered that an identical coloration of regions between maps is irrelevant. What is relevant is the comparison of the shape of the detected regions between the different maps, i.e. which pixels within a single map are clustered and thus colored identically, and how the maps convey color contrast and perceived color distance between their segmented regions.

To evaluate the presented segmentation methods (H^2SOM and serial coupling of dimension reduction and clustering) both were applied to the barley seed data set ($\mathcal{I}^{\mathrm{B}}$).

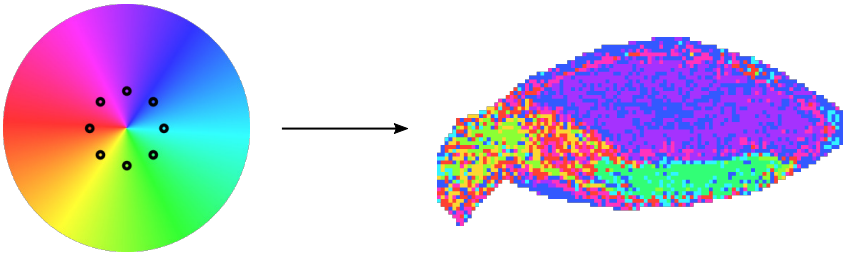
H^2SOM results

Figure 5.6 shows the result of the H^2SOM algorithm with $\mathfrak{R} = 3$ rings, $\mathfrak{n} = 8$ neighbors, a learning rate of $\varepsilon = 1.1$ with a decay of 0.1 and a neighborhood bell function with of $\mathfrak{w} = 12$ and a decay of 1.
Figure 5.6A presents the color disc projection of the first ring and the associated segmentation map. A comparison to the morphological structure of the barley seed (compare Figure 2.1) shows that endosperm and shoot are well reconstructed. The center is also recognizable, but it shares a cluster with several pixels around the outer edge of the seed. The root is slightly recognizable, but only as a union of a multitude of clusters, which results in a variety of different colors, making this

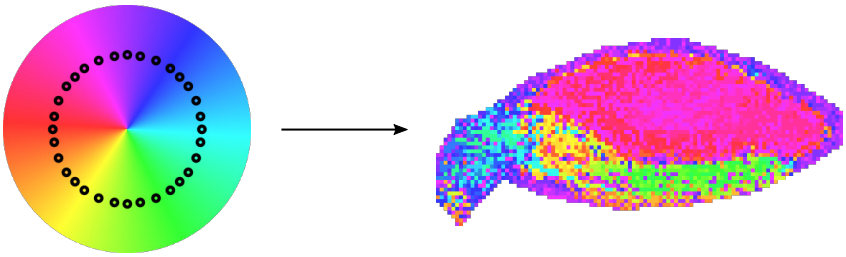
region less well defined than the others. The scutellum is not visible at all.
Figure 5.6B presents the clustering according to the second ring. It can be seen that the separation of the different regions is improved by increased color contrast. The root is better recognizable because it is assigned to the blue and cyan prototypes, while most of the cyan pixels of the shoot region from the first ring are now assigned to the colors green, yellow or orange. The center is also better recognizable. The pixels are mostly assigned to orange and yellow colors, which provides a good color contrast to the adjacent morphological structures. Furthermore, the outer edge of the seed, which is most likely matrix area, is now better separable from the endosperm.
Figure 5.6C presents the clustering according to the third ring. It appears that for some regions the color contrast is increased, such as the shoot and the center, while it is decreased for others, such as the root. The morphological regions are also less homogeneous in color. This can probably be explained by a low inherent dimensionality of the barley seed. Therefore, the number of prototypes in the third ring may overestimate the number of inherent classes. Consequently, many clusters will remain empty or consist of only a few members, which can lead to a decrease in color homogeneity and the introduction of artificial classes. The decreased homogeneity becomes visible in the endosperm and the shoot. The problem of artificial classes is possibly visible in the root, through the differentiation of the purple and green regions. However, this cannot be claimed with certainty, since the root area, in particular, consists only in small parts of tissue and to a large extent of matrix. So it could be correct that the yellow, green and orange pixels belong to the matrix, like the outer edge of the seed, whereas the purple pixels belong to the tissue.

Position optimization results To evaluate the position optimization algorithms both variants ("winner-takes-all" and "tug-of-war") are applied to the first ring of the H^2SOM described above. The result is shown in Figure 5.6. To illustrate the course of position changes of the neurons through the successive iterations until convergence, the new positions after two iterations and after convergence by reaching the automatic stop criterion are shown. In addition, Table 5.1 presents the number of iterations required to converge for each algorithmic variant.
Figure 5.6 shows that the "winner-takes-all" variant has a higher change rate per iteration, which explains the faster convergence shown in Table 5.1. It can be seen that the color contrast for the "winner-takes-all" segmentation map after two iterations is notably better and allows a more accurate distinction between the different regions. This is particularly apparent through improved color homogeneity

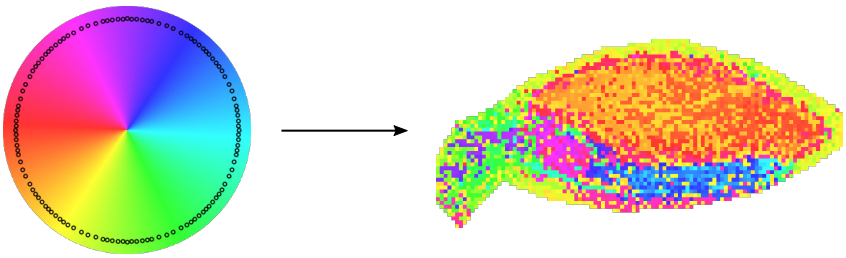
for the endosperm and the shoot and a decreased perceived distance between the colors of the root and center region. For the "tug-of-war" variant, it can be seen that due to the small change in position the colors change barely. After convergence, both variants lead to comparable results. However, the “winner-takes-all” variant provides a better distinction between the shoot and the center-root area. The results in “mitigated” mode are very similar, which is why they are not presented.



(A) First ring



(B) Second ring



(C) Third ring

Fig. 5.6.: Application of the H^{2}SOM algorithm to the barley data set $\mathcal{I}^{\mathrm{B}}$ – H^{2}SOM parameters: $\mathfrak{n} = 8$, $\varepsilon = 1.1$ with decay $= 0.1$, $\mathfrak{w} = 12$ with decay $= 1$ and $\mathfrak{R} = 3$. The left side shows the projection on the color disc and the right side presents the associated segmentation map.

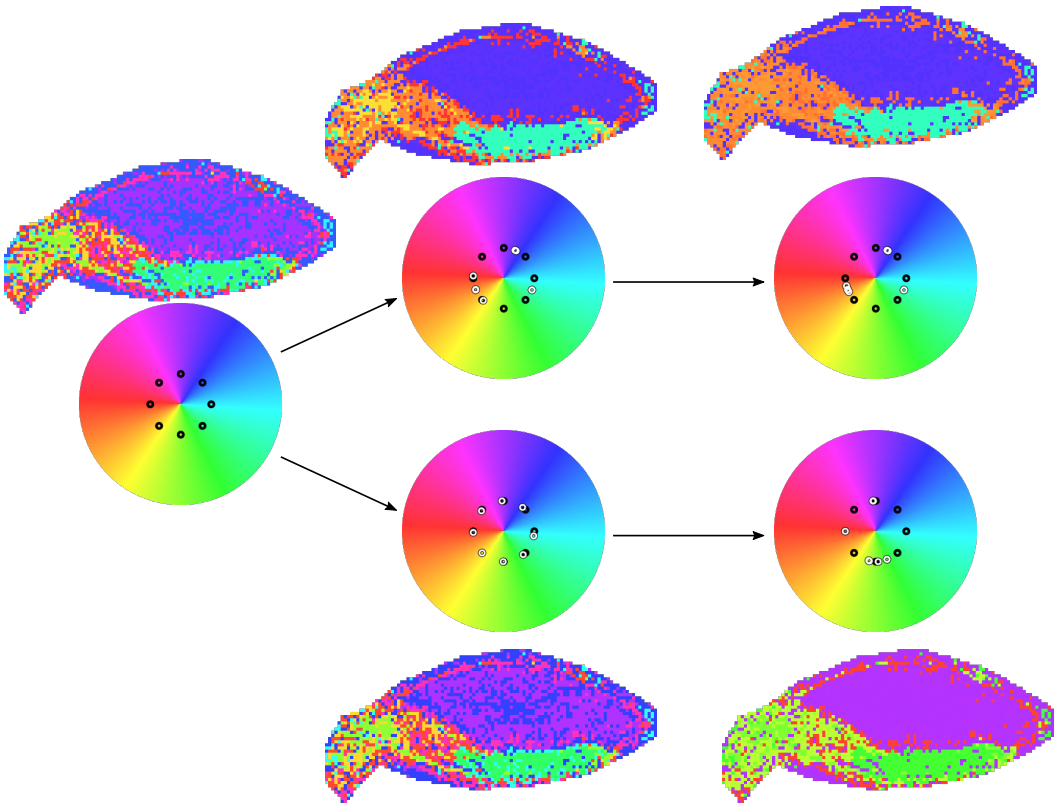


Fig. 5.7.: Position optimization application example – Application of the two position optimization methods, "winner-takes-all" and "tug-of-war". Both methods were executed without "mitigated" mode. The left combination of color disc projection and segmentation map represents the first ring of an H^2SOM with parameters equivalent to Figure 5.6. The middle combinations illustrate the position changes (original position: black, new position: white) of the neurons using the "winner-takes-all" method (top) and the "tug-of-war" method (bottom) after two iterations. The right combinations illustrate the position changes after reaching the automatic stop criterion.

Embedding projection results

The computation of segmentation maps through clustering using different setups of the "embed-first" and "cluster-first" methods are evaluated in Figures 5.8 to 5.11. If not stated otherwise, all experiments use a k-Means clustering with eight clusters.

Influence of the lightness encoding Figure 5.8 shows segmentation maps for the three presented algorithmic variants ("embed-first", "cluster-first" and "embedding projection") to demonstrate the importance of encoding the distance of a prototype from the center using the lightness of the HSL color space. For each variant, the segmentation map with lightness encoding (Figure 5.8A,C,D) offers a better color contrast and facilitates the differentiation of different regions. This becomes

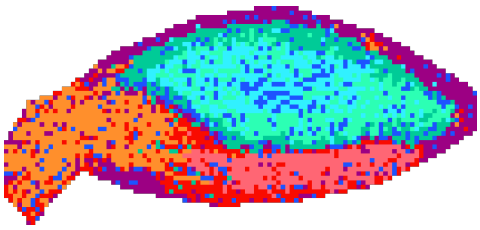
Tab. 5.1.: Number of iterations for ring-wise position optimization – Number of iterations until the automatic stop criterion was reached. Data set: $\mathcal{I}^{\mathrm{B}}$, H^{2}SOM parameters are equivalent to Figure 5.6. "wta": "winner-takes-all", "wtam": "winner-takes-all-mitigated", "tow": "tug-of-war", "towm": "tug-of-war-mitigated".

Ring	"wta"	"wtam"	"tow"	"towm"
First	6	8	20	24
Second	10	11	129	173
Third	11	11	2241	2309

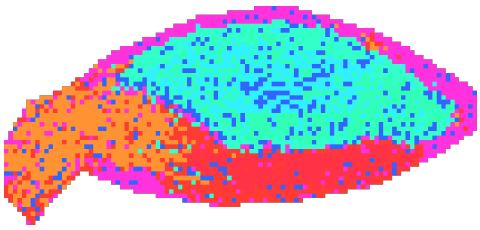
particularly clear when comparing the shoot (red) and endosperm (cyan) regions in Figure 5.8A,B and Figure 5.8E,F. In both cases, these two regions can be divided into an inner and an outer region. In Figure 5.8A,E this is expressed by a lighter color for the inner region and a darker color for the outer region. Compared to the H^{2}SOM approach, the UMAP "embed-first" and "embedding projection" segmentation maps are quite similar to the maps computed with the H^{2}SOM after position optimization with the converged "winner-takes-all" variant. The segmentation map of the UMAP "cluster-first" approach is more similar to the H^{2}SOM without position optimization, with an improved segmentation of the root region and a stronger subdivision in the shoot region.

Influence of the dimension reduction method Different dimension reduction methods can have a strong influence on the clustering and thus on the resulting segmentation map. This is demonstrated in Figure 5.9. Compared to the PCA results (B, D, F), the UMAP results (A, C, E) show segmentation maps with more homogeneous and more clearly defined regions. Most affected by this observation is the "embed-first" variant, followed by the "embedding projection" and the "cluster-first" variant. The reason for the better results with UMAP is probably that UMAP can preserve the topology of the original feature space better than PCA for most data sets. Consequently, clusters resulting from a distance- or density-based clustering methods applied to a UMAP embedding will be closer to the clusters in the original space. A similar argument applies to the projection of the prototypes. The relationship between them in a UMAP embedded space will be closer to the relationship in the original space than in a PCA embedded space, provided that UMAP computes a well-preserved topology.

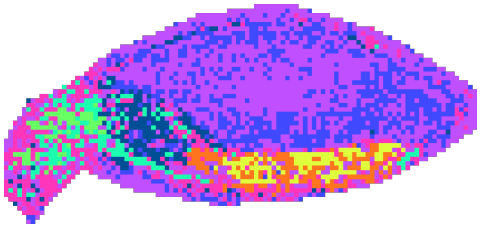
Influence of the clustering method Similar to the applied dimension reduction method, the applied clustering method can have a strong influence on the segmentation result. This is demonstrated in Figure 5.10 using k-Means clustering,



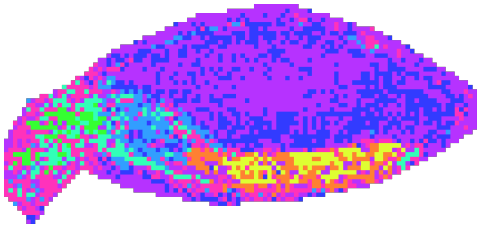
(A) UMAP “embed-first” encoded lightness



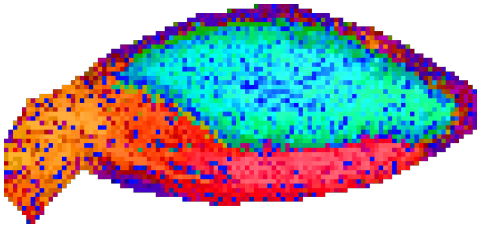
(B) UMAP “embed-first” constant lightness



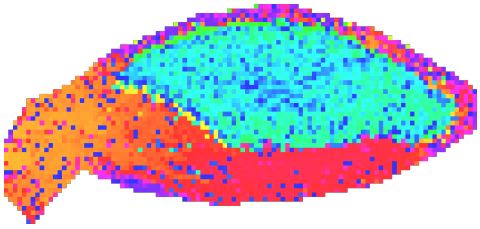
(C) UMAP “cluster-first” encoded lightness



(D) UMAP “cluster-first” constant lightness



(E) UMAP full embedding encoded lightness

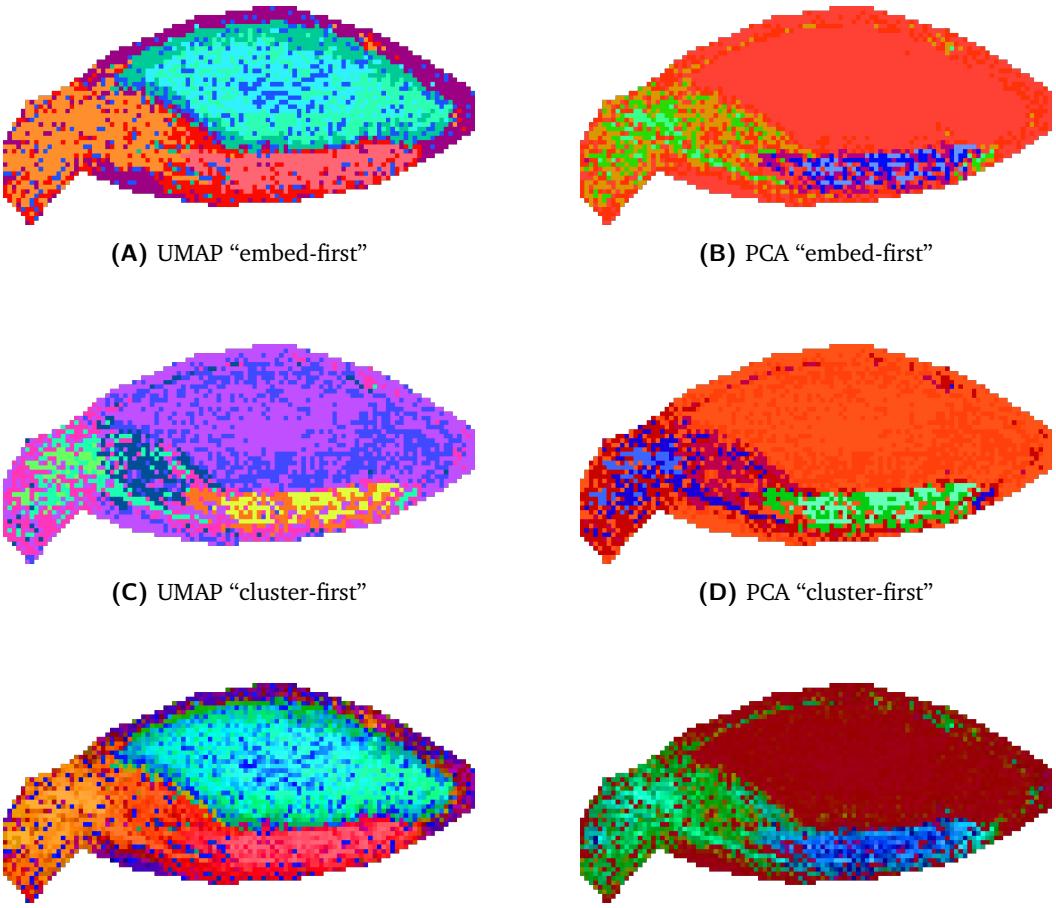


(F) UMAP full embedding constant lightness

Fig. 5.8.: Lightness effect – Segmentation maps of the barley seed data set. All maps were generated with UMAP as dimension reduction method, k-Means as clustering method and eight clusters. The first row (A,B) shows the “embed-first” approach, the second row (C,D) shows the “cluster-first” approach and the third row (E,F) shows the projection of all embedded data points. The left column (A,C,E) encodes the distances of all prototypes from the center with the lightness of the HSL color space. Each distance value is linearly scaled using the boundary mapping $[0, 1] \mapsto [0.25, 0.75]$. The right column (B,D,F) uses a constant lightness.

agglomerative hierarchical ward clustering and DBSCAN in combination with UMAP and the two variants “embed-first” and “cluster-first”. For k-Means and agglomerative hierarchical ward clustering, the number of clusters was set to eight to remain comparable with previous results. DBSCAN is a method that automatically determines the number of clusters. For the UMAP “embed-first” approach, DBSCAN resulted in ten clusters. For the UMAP “cluster-first” approach DBSCAN resulted in one single cluster, which means that no clusters were found and no segmentation map could be computed.

A direct comparison between Figure 5.10A and Figure 5.10C shows that the com-



(A) UMAP "embed-first"

(B) PCA "embed-first"

(C) UMAP "cluster-first"

(D) PCA "cluster-first"

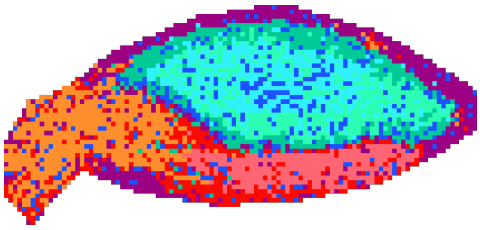
(E) UMAP full embedding

(F) PCA full embedding

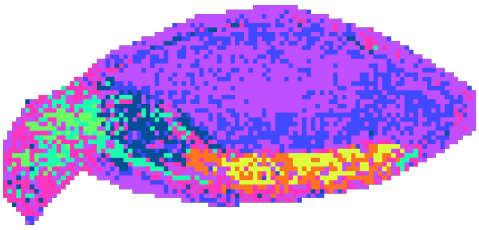
Fig. 5.9.: Embedding effect – Segmentation maps of the barley seed data. All maps were generated using the lightness encoding, k-Means as clustering method and eight clusters. The first row (A,B) shows the "embed-first" approach, the second row (C,D) shows the "cluster-first" approach and the third row (E,F) shows the projection of all embedded data points. The left column (A,C,E) was computed using UMAP for dimension reduction. The right column (B,D,F) was computed using PCA for dimension reduction.

puted segmentation maps are very similar. Only small areas, such as a part between the center and the shoot, or individual pixels are different. A comparison between Figure 5.10B and Figure 5.10D shows a similar tendency, considering that the exact colors are not important. Although the differences are more noticeable, e.g. in root and shoot, the rough definition of the regions is still similar. This is especially the case when the union of two clusters is considered, e.g. similar definitions of the shoot region (union of the yellow and orange clusters in Figure 5.10B and union of the bluish clusters in Figure 5.10D) and the endosperm (union of purple and blue clusters in Figure 5.10B and union of the orange and pinky clusters in Figure 5.10D). The clustering with DBSCAN in Figure 5.10E, on the other side,

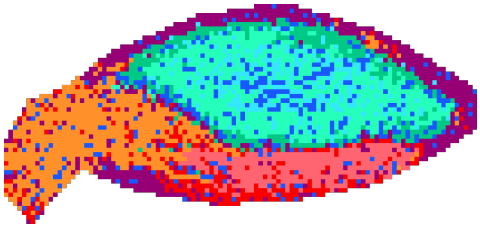
shows much clearer differences. The main regions are similar to Figure 5.10A and Figure 5.10C, i.e. endosperm, shoot, a union of center and root and the outer edge of the seed (presumably matrix). However, the endosperm and shoot appear to be way more homogeneous using DBSCAN. Additionally, there is a distinct yellow-greenish cluster consisting of only a few pixels close to the scutellum. Without further biological examination, the differences in quality between the segmentation maps of Figure 5.10A,C and Figure 5.10E cannot be assessed. While homogeneous regions may be a desired result, it may also obscure layered substructures that could be part of the sample morphology.



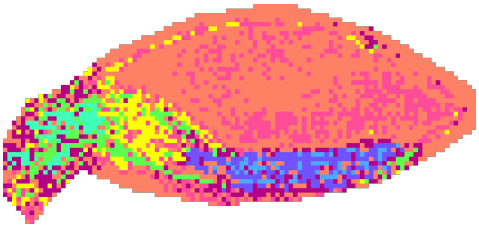
(A) UMAP "embed-first" k-Means



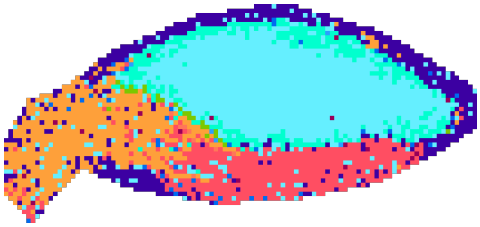
(B) UMAP "cluster-first" k-Means



(C) UMAP "embed-first" Ward clustering



(D) UMAP "cluster-first" Ward clustering



(E) UMAP "embed-first" DBSCAN

Fig. 5.10.: Effect of the clustering method – Segmentation maps of the barley seed data set. The maps were generated with UMAP as dimension reduction method, lightness encoding and different clustering methods. Ward clustering is used as an abbreviation for agglomerative hierarchical clustering with Euclidean distance and Ward's method as the linkage criterion. For k-Means and Ward clustering the number of clusters was set to eight. DBSCAN automatically sets the number of clusters to ten. The combination of DBSCAN and "cluster-first" results in one single cluster, i.e. no segmentation.

Influence of the serial application of dimension reduction methods The last evaluation addresses the approach to apply two dimension reduction methods in succession to improve the computation time. For this purpose, the urinary mouse bladder data set $\mathcal{I}^{\mathrm{U}}$ after matrix subtraction and before peak picking is used. This data set has a dimensionality of 8562 features, while the barley data set is limited to 101 features. The evaluation will focus on the methods PCA and UMAP.
Figure 5.9 already indicated that UMAP is the more effective method for computing segmentation maps with the presented algorithmic approaches. However, concerning computing time, PCA is the more efficient method. To combine the advantages of both methods, we proposed to compute a pre-embedding using PCA, followed by the computation of the two-dimensional embedding using UMAP.
To get an impression of how a pre-embedding influences the result, five different approaches are presented. Three of these approaches reduce the feature space through pre-embeddings:

1. A direct UMAP embedding ($\mathbb{R}^{8562} \xrightarrow{\text{UMAP}} \mathbb{R}^2$).
2. A pre-embedding reduction to 1000 features ($\mathbb{R}^{8562} \xrightarrow{\text{PCA}} \mathbb{R}^{1000} \xrightarrow{\text{UMAP}} \mathbb{R}^2$).
3. A pre-embedding reduction to 100 features ($\mathbb{R}^{8562} \xrightarrow{\text{PCA}} \mathbb{R}^{100} \xrightarrow{\text{UMAP}} \mathbb{R}^2$).
4. A pre-embedding reduction to 10 features ($\mathbb{R}^{8562} \xrightarrow{\text{PCA}} \mathbb{R}^{10} \xrightarrow{\text{UMAP}} \mathbb{R}^2$).
5. A direct PCA embedding ($\mathbb{R}^{8562} \xrightarrow{\text{PCA}} \mathbb{R}^2$).

Table 5.2 compares the computation time of the five different approaches. As mentioned above, PCA is considerably faster than UMAP. Interestingly, the pre-embedding of $\mathbb{R}^{8562} \rightarrow \mathbb{R}^{1000}$ is slower than a direct computation with UMAP. We assume that a UMAP embedding of $\mathbb{R}^{1000} \rightarrow \mathbb{R}^2$ is not much more efficient in terms of computation time than $\mathbb{R}^{8562} \rightarrow \mathbb{R}^2$. Due to the overhead of computing two embeddings, this may result in higher computation time for the pre-embedding. However, the other two pre-embedding approaches result in faster computation times. This shows that the pre-embedding approach is not always suitable, but with a sufficiently large dimensionality of the original space and a sufficiently strong reduction, the computation time can be considerably reduced.
To investigate the influence of pre-embeddings on the segmentation results, the three segmentation variants “embed-first”, “cluster-first” and embedding projection are computed for all five approaches. The results are presented in Figure 5.11. The direct embeddings with UMAP (Figure 5.11A,B,C) and PCA (Figure 5.11N,M,O) ($\mathbb{R}^{8562} \rightarrow \mathbb{R}^2$) can be considered as reference.

Interestingly, no major differences in quality are visible between the direct embedding with UMAP and the pre-embedding variants are visible, considering that the exact colors are not important. All nine pre-embedding approaches are capable to distinguish the urothelium from the surrounding tissue areas, such as the detrusor muscle and the lamina propria (compare Figure 2.7) to a similar extent. While the mixed clusters in the area of the detrusor muscle and the lamina propria in Figure 5.11J appear somewhat more heterogeneous, this is mainly due to a more strongly perceived color difference at the red to the orange color border. To support this claim, Figure 5.12 presents a comparison of the segmentation map shown in Figure 5.11J with a rotated color disc. The rotation shifts the clusters of the detrusor muscle and the lamina propria into the bluish region, which reduces the color contrast and makes these segmented regions appear more homogeneous. Nevertheless, all morphological characteristics that are visible in Figure 5.11J are preserved. Moreover, all nine approaches assign the tissue cuts in the inner part of the urothelium to the same cluster as the outer matrix area and can detect the adventitial layer to varying extents. Six of the nine segmentation maps, i.e. the "embedding-first" and "embedding projection" methods (Figure 5.11A,C,D,F,G,I,J,L), show a stronger differentiation between two matrix regions. However, this observation does not depend on the embedding procedures and is therefore negligible.

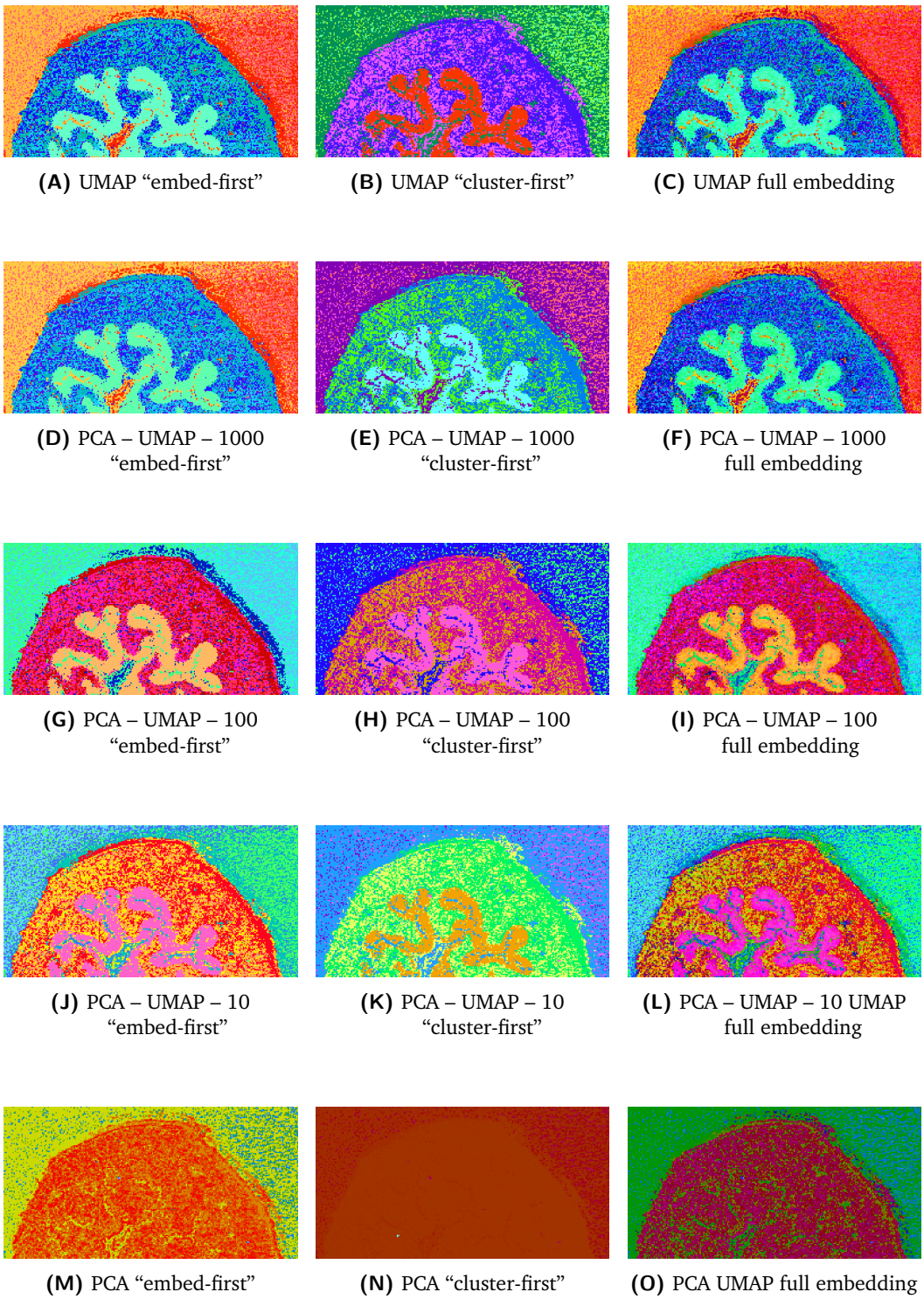
As with the barley seed data set (compare Figure 5.9), the segmentation maps computed using a direct embedding by PCA show substantially worse results. Each of the three maps (Figure 5.11M,N,O) only indicates a differentiation between tissue and matrix, whereas Figure 5.11N shows a considerably worse contrast that is hardly perceptible.

The results of the mouse urinary bladder data set indicate that a pre-embedding does not significantly change the quality of the segmentation result. To explain and generalize this observation, we make the hypothesis that there might be a relation between the dimensionality of the pre-embedding and the intrinsic dimensionality of the data. The mouse urinary bladder is well structured. Thus, it can be assumed that it has a low intrinsic dimensionality so that UMAP can preserve the topology of the original feature space even with a pre-embedding to ten features. However, for a data set with higher intrinsic dimensionality, a pre-embedding that provides too few features might cause problems. Consequently, a sufficient number of features should be selected for the pre-embedding depending on the data set. However, further research is required to substantiate this assumption.

Tab. 5.2.: Pre-embedding computation time comparison – Computation time comparison of different dimension reduction approaches for the mouse urinary bladder data set $\mathcal{I}^{\mathrm{U}}$. UMAP and PCA indicate direct embeddings ($\mathbb{R}^{8562} \xrightarrow{\text{UMAP}} \mathbb{R}^2$ and $\mathbb{R}^{8562} \xrightarrow{\text{PCA}} \mathbb{R}^2$), while 1000, 100 and 10 indicate the use of pre-embeddings ($\mathbb{R}^{8562} \xrightarrow{\text{PCA}} \mathbb{R}^{1000/100/10} \xrightarrow{\text{UMAP}} \mathbb{R}^2$).

Statistic	UMAP	1000	100	10	PCA
Mean	108.422s	175.097s	71.074s	55.201s	13.232s
Std	5.519s	4.905s	0.774s	0.961s	0.428s

Summary In summary, all presented methods (H^2SOM, "embed-first", "cluster first" and "embedding projection") resulted in high-quality segmentation maps for the given examples if parameterized accordingly.



(A) UMAP "embed-first" **(B)** UMAP "cluster-first" **(C)** UMAP full embedding

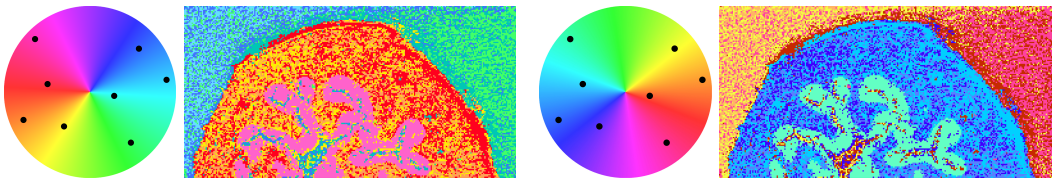
(D) PCA – UMAP – 1000 "embed-first" **(E)** PCA – UMAP – 1000 "cluster-first" **(F)** PCA – UMAP – 1000 full embedding

(G) PCA – UMAP – 100 "embed-first" **(H)** PCA – UMAP – 100 "cluster-first" **(I)** PCA – UMAP – 100 full embedding

(J) PCA – UMAP – 10 "embed-first" **(K)** PCA – UMAP – 10 "cluster-first" **(L)** PCA – UMAP – 10 UMAP full embedding

(M) PCA "embed-first" **(N)** PCA "cluster-first" **(O)** PCA UMAP full embedding

Fig. 5.11.: Effect of serially coupled dimensional reduction – Segmentation maps of the mouse urinary bladder data set $\mathcal{I}^{\mathrm{U}}$. The three columns refer to the three variants: "embed-first", "cluster-first" and "embedding projection". The five rows refer to different dimension reduction approaches: 1. UMAP ($\mathbb{R}^{8562} \xrightarrow{\text{UMAP}} \mathbb{R}^2$), 2. PCA serially coupled with UMAP ($\mathbb{R}^{8562} \xrightarrow{\text{PCA}} \mathbb{R}^{1000} \xrightarrow{\text{UMAP}} \mathbb{R}^2$), 3. PCA serially coupled with UMAP ($\mathbb{R}^{8562} \xrightarrow{\text{PCA}} \mathbb{R}^{100} \xrightarrow{\text{UMAP}} \mathbb{R}^2$), 4. PCA serially coupled with UMAP ($\mathbb{R}^{8562} \xrightarrow{\text{PCA}} \mathbb{R}^{10} \xrightarrow{\text{UMAP}} \mathbb{R}^2$), 5. PCA ($\mathbb{R}^{8562} \xrightarrow{\text{PCA}} \mathbb{R}^2$).



(A) Projection as in Figure 5.11J **(B)** Counterclockwise rotation by 168°

Fig. 5.12.: Color contrast improvement – Illustration how the rotation of the color disc influences the color contrast. (A) Clusters of the detrusor muscle and the lamina propria region appear heterogeneous. (B) The same region appears more homogeneous while retaining all morphological characteristics that can be seen in (A).

5.4 Interactive Visual Exploration of Molecular Composition based Segmentation Maps

To enhance and extend the analysis capabilities of molecular composition based segmentation maps through interactive visual exploration, we have developed a visual analysis (web)tool, called WHIDE (**w**eb-based **h**yperbolic **i**mage **d**ata **e**xplorer). WHIDE is designed to allow a dynamic and interactive exploration segmentation maps utilizing color disc projections and to complement these maps with additional information. The first version of WHIDE was developed by Jan Kölling and Daniel Langenkämper [54], with a general focus on multivariate bioimages and segmentation maps computed with the H^2SOM algorithm.
This thesis presents the second version of WHIDE. WHIDE 2.0 is a completely revised, redesigned and re-implemented version. The implementation was done in cooperation with Jonas Bicker as part of a student project and a bachelor thesis. Under my supervision, he implemented major parts of the frontend and backend, as well as parts of the offline component. Conception, development and design, minor parts of the implementation in backend and frontend, as well as major parts of the implementation of the offline components were done by me.

The majority of this section will focus on the use of WHIDE in combination with the H^2SOM algorithm. A discussion about the use in combination with other clustering methods follows at the end.

Interface

Following the design guidelines defined in Section 1.4, WHIDE kept very simplistic. Most parts of the user interface are presented in Figure 5.13. The interface consists of four main visual building blocks which are referred to as displays:

1. **The segmentation map display**, which contains the computed segmentation map. If provided, it can also contain another imaging modality, which is used as a background.

2. **The color disc display**, which contains the HSL color disc projection, i.e. the HSL color disc and the projected prototypes, which are displayed as circles.

3. **The spectra display** (labeled as bookmarks), which contains selected CIPRA glyphs (explained below).

4. **The list display**, which contains all m/z-values present in the data set.

To avoid unnecessary option menus, all displays are equipped with fast-access controls. This way, all necessary functionalities are immediately visible and accessible to the user.

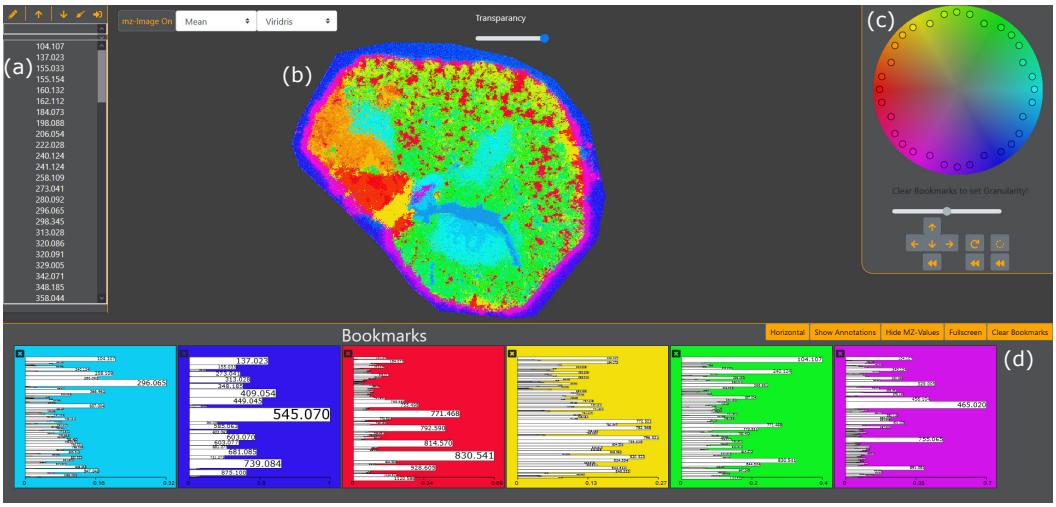


Fig. 5.13.: WHIDE interface – Overview of the WHIDE user interface. The segmentation map shows the clustering result of the second ring of an H^2SOM trained with the mouse kidney data set $\mathcal{I}^K$. H^2SOM parameters: $\mathfrak{n} = 8$, $\varepsilon = 1.1$ with decay $= 0.01$, $\mathfrak{w} = 120$ with decay $= 1$ and $\mathfrak{R} = 3$. (a) List display; (b) segmentation map display; (c) HSL color disc display; (d) spectra display (bookmarks) with active CIPRA glyphs.

Cluster exploration

An essential aspect of effective and efficient knowledge discovery in complex and high-dimensional data, using visual analysis tools, is the use of multiple, complementary and interconnected visualizations (visual elements). Therefore, all four displays are linked and complement each other with information.

In order to extract information from segmentation maps, one of the first and primary steps is to examine the individual clusters. Various properties are of interest, such as the spatial distribution and the cohesion of the clusters. To facilitate the spatial analysis of individual clusters, the segmentation map display offers a highlighting functionality. Moving the cursor across the segmentation map causes the entire cluster of the currently selected pixel to be colored white. The color white was chosen intentionally as it is excluded from the possible segmentation colors. Thus, the highlighting functionality does not collide with the colors of the segmentation map.

With an increasing number of clusters comes an increasing number of prototypes and colors. The result is a denser population of the color disc and a decrease of color contrast. To provide an overview of the position of the prototype on the color disc, not only the pixels of the focused cluster on the segmentation map are highlighted in white, but also the corresponding prototype on the color disc. Vice versa, this also applies to hovering a prototype, i.e. individual clusters can be highlighted by hovering a prototype. An increasing number of prototypes will also lead to the formation of smaller clusters. To facilitate the investigation of clusters of varying sizes, a zoom functionality is provided. For navigation in a zoomed state, the segmentation map can be moved freely using a drag and drop interaction. These functionalities allow easy handling of segmentation maps of various sizes and focusing on individual regions in varying levels of detail.

Combinatorial intensity profile archetype

Another aspect of interest is the analysis of the feature domain of each cluster. In the case of MSI data and multivariate spectral data in general, the feature domain refers to the measured intensities in the mass spectra. However, the analysis of each individual spectrum of a cluster is a rather inefficient way to explore the feature domain of a cluster. A more efficient way to get an overview of the feature domain of a cluster is to analyze the features of the prototypes associated with each cluster. Just like the mass spectra of the input data, each prototype has a numerical values associated with each m/z-value. These values are called coefficients. In contrast to the intensities of the mass spectra, the coefficients of a prototype are not measured but calculated from the input data. The coefficient values can be interpreted as the weighting (or relevance) of the respective features (m/z-values) for the corresponding cluster. WHIDE covers the analysis of the feature domain of clusters with CIPRA glyphs. A CIPRA glyph (**c**ombinatorial **i**ntensity **pr**ofile **a**rchetype) is an abstracted graphical representation of the feature domain. The implementation is based on the original work on WHIDE [54].

In general, a glyph is a graphical representation that maps a set of data features to different graphical attributes. Examples of such attributes are size, shape and color. Examples of graphical representations are Chernoff faces [20], star glyphs [19], color icons [60] or stick figures [91]. The CIPRA glyph combines aspects of a bar chart, a star glyph and is partly inspired by the sequence logo visualization, which is used to represent patterns in nucleotide or amino acid sequences [104]. In a sequence logo, the feature values (i.e. characters) for each position of a set of aligned sequences are arranged on top of each other and sorted according to their

frequency. In the visual representation, the height of the characters represents their frequency at the respective position. With this visualization, prominent sequence patterns can be quickly identified because they are more prominently visible in the logo.

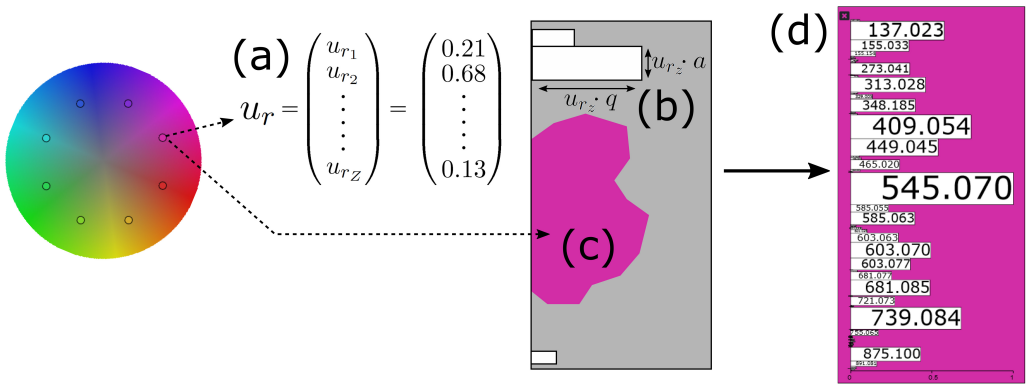


Fig. 5.14.: CIPRA glyph creation – For each prototype of the trained H^{2}SOM, the prototype coefficients (a) are visualized as bars. The width and height of each bar are proportional to the coefficient value and are scaled by constant width q and height m factors (b). The background of the glyph depends on the position of the prototype on the HSL color disc (c). Additionally, the m/z-value associated with each coefficient is shown within the bar (can be disabled). A final CIPRA glyph is illustrated by (d).

A similar approach is applied for the construction of the CIPRA glyphs, which is illustrated in Figure 5.14. For a selected prototype of a trained H^{2}SOM a CIPRA glyph is constructed by visualizing the coefficients as a modified bar chart. The width and height of each bar are proportional to the coefficient value and are scaled by constant width and height factors. In addition, the information presented by each bar is expanded by adding the m/z-value associated with each coefficient as text within the bar. The size of the text is proportional to the size of the bar. According to the "details-on-demand" approach, as presented in Section 1.4, the text is disabled by default and must be enabled manually. To selectively have access to the m/z-value information while the text is disabled, an annotation label is activated on hover. The numerical value of a coefficient, expressed as the size of the bar, indicates the influence of the corresponding m/z-value on the structure of the cluster. With the described triple encoding (height, width, text size), the most influential m/z-values of a cluster can be quickly identified.

To clearly show which CIPRA belongs to which prototype, the background color corresponds to the color of the color disc projection of the prototype. This not only enables a quick and easy visual mapping between a CIPRA and a prototype but also to the corresponding cluster in the segmentation map. In addition, every activated

CIPRA responds to the highlighting functionality, just like the clusters and prototypes. This is realized by showing a white border.

Furthermore, a bottom axis is used to allow a rough comparison between the CIPRAs. This axis shows the value of the largest coefficient, which helps to rank the size of the bars between the CIPRAs. According to the "zoom and filter" approach, all CIPRAs are deactivated by default and are displayed after manual selection by selecting either a prototype or a pixel of the segmentation map. Using CIPRA glyphs to visualize prototypes reduces the complexity of the data considerably, which is beneficial for the visual presentation. However, the CIPRA glyph still conveys enough information so that the most important features of the feature domain of each cluster can be visually explored. If interesting CIPRAs are found, the associated mass spectra of the respective cluster can be analyzed in detail using other tools.

As the number of features increases, the presented vertically organized CIPRA glyph can become cluttered. To solve this problem, a horizontal version can be used. This provides more space for the glyph presentation since most display devices use a landscape format, such as most computer monitors. The horizontal mode also offers a visualization of the coefficients in centroid mode, which can be interpreted as an artificial prototype spectrum. This visual representation should be more familiar to users who regularly work with spectral data, which may make it easier to understand the CIPRA glyph. All three visualizations are illustrated in Figure 5.15.

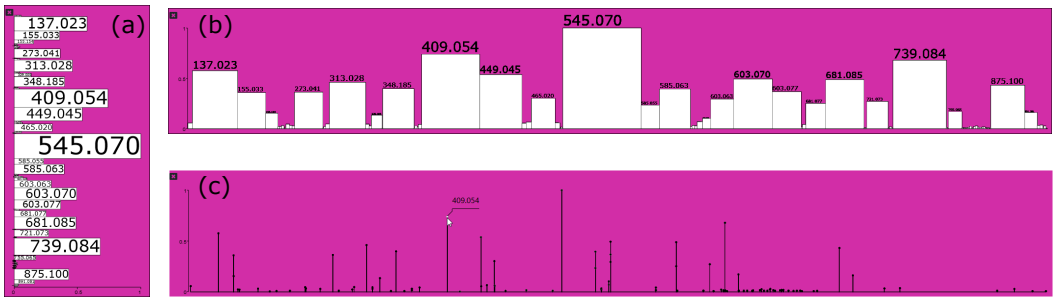


Fig. 5.15.: Comparison of prototype coefficient visualizations – The coefficients of a prototype can be displayed either as vertical (a) or horizontal (b) CIPRA glyph or as an artificial spectrum in centroid mode (c).

Color contrast

When using a categorical colormapping to visualize segmentation maps, there is no way to guarantee the optimal color contrast for each cluster or part of the segmentation map. This is where the combination of HSL color disc projection and

the interactivity provided by WHIDE comes to the rescue. The color disc display offers two transformation functionalities to address this problem.

The first way to transform the prototype projection is to change the focus point of the unit circle projection. This is done by translating the focus using Möbius transformations as discussed in Section 5.3.1. Due to the strong "fish-eye" effect, prototypes that are closer to the focus will spread more widely across the circle, while prototypes that are further apart will be squeezed together at the edge. This has two effects: first, the color contrast can be specifically increased for a subset of prototypes, which facilitates differentiation and also allows a stronger focus on certain differences, and second, the color contrast can be reduced for a subset of prototypes, which gives an impression of the effect of potential cluster merges.

The second way to transform the prototype projection is by rotation. In contrast to the other transformation, the position of the prototypes remains the same, but the colors are rotated. The order of the colors within the color circle remains untouched. It may not be immediately obvious, but this technique can also be used to increase color contrast for certain clusters while reducing it for others. The reason for this is that the perceived distances in the HSL color space are not uniform, which was discussed in Section 5.3.1. Consequently, certain color transitions create a higher perceived color contrast, e.g. the human eye has a lower sensitivity for blue colors than for red or green colors [47].

In order to be able to track the color changes and maintain coherence, the coloring of the segmentation map as well as of all active CIPRAS is changed on the fly during the application of either of the transformations. This makes the coloring of the visual displays dynamic and responsive.
An example of how both transformations can be used to enhance the contrast of certain aspects of clustering, while intentionally reducing it for others, is presented in Figure 5.16. In this example, the intention is to improve the color contrast between the clusters within the cortex and medulla region, while reducing it for the matrix region. It can be seen that with the coloring of the original projection the different clusters within the cortex (green, cyan) and the medulla (orange) region are difficult to distinguish. By using rotation to shift the prototypes into a different color range (purple, red, orange,), the differentiation is already easier (see Figure 5.16b). A shift of the focus that squeezes the prototypes of the matrix area closer together (see Figure 5.16c) facilitates cluster differentiation within the cortex and medulla regions even more. However, a combination of both transformations produces even better color contrasts for the clusters within both regions, while the color contrast for clusters of the matrix region is reduced (see Figure 5.16d).

This small example demonstrates that the ability to adjust colors dynamically and interactively is very beneficial to improve the differentiation of certain clusters and to direct the focus of attention. This would not be possible with a static visualization.

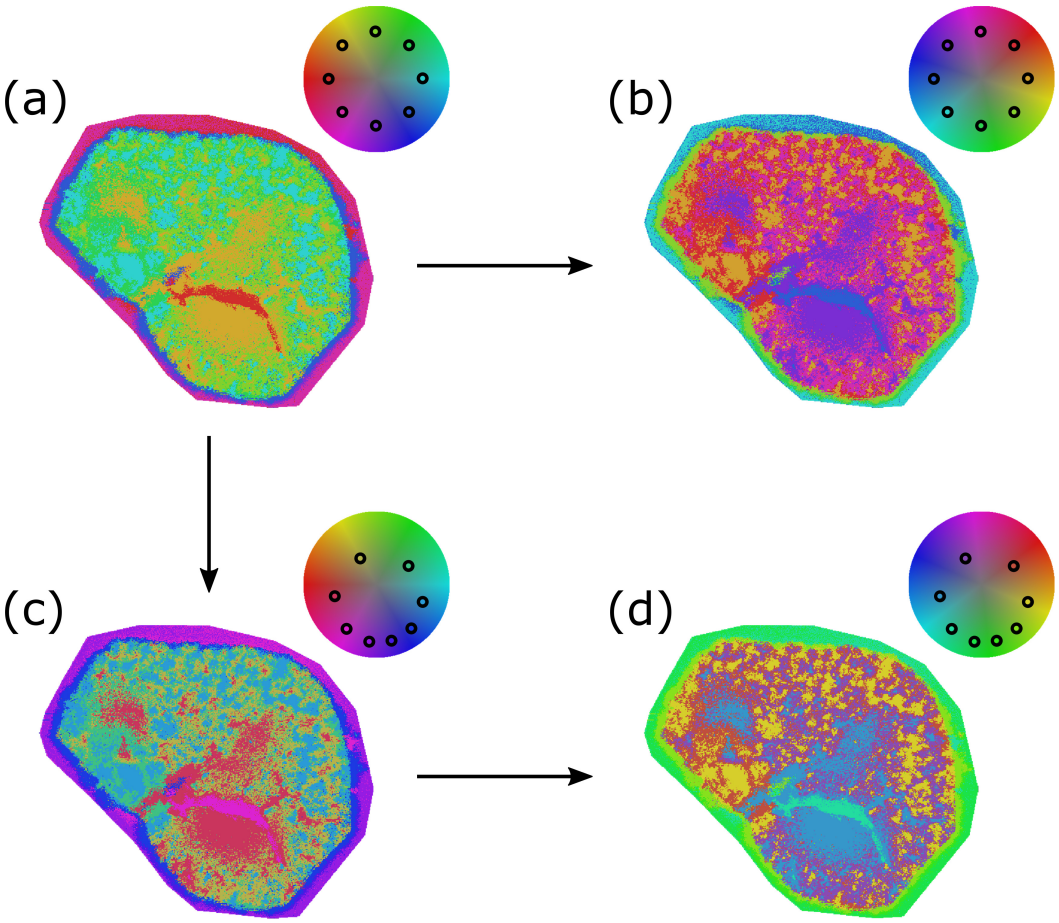


Fig. 5.16.: Interactive HSL color disc transformations – Segmentation maps of the first ring of an H^{2}SOM trained with the mouse kidney data set $\mathcal{I}^{\mathrm{K}}$. H^{2}SOM parameters are equivalent to Figure 5.13. The four segmentation maps show the coloration with the initial prototype projection result (a), coloration after transformation using rotation (b), coloration after transformation using focus translation (c) and coloration after a combined transformation of focus translation and rotation (d).

H^{2}SOM hierarchies Another interaction functionality provided by the color disc display uses the hierarchical organization of the H^{2}SOM. It enables the user to dynamically switch back and forth between the individual rings of the H^{2}SOM. Changing the number of clusters by switching between the rings allows the examination of segmentation results with different granularity.

Annotations

To be able to incorporate knowledge about identified m/z-values, the list display offers an annotation functionality. Once an annotation has been made, it is not only visible in the list display but can also be dynamically activated and deactivated in the CIPRA glyph visualization.

Overlays

The segmentation map display is organized as a layer-based system (illustrated in Figure 5.17). That way it provides various options for superimpositions.

Highlight overlay The function of the highlighting overlay is to highlight all pixels of a selected cluster. The highlighting is done by coloring the corresponding pixels in white. A cluster can be selected by selecting a pixel in the segmentation map, a prototype in the color disc display or a CIPRA. An example is shown in Figure 5.18D.

Modality overlay Another overlay option is provided by the modality overlay. If a pre-aligned image modality is loaded, it is placed at the lowest level of the segmentation map display. This causes the segmentation map to automatically superimpose it. By default, nothing of the modality image will be visible. A continuous transparency slider can be used to dynamically adjust the transparency of the segmentation map. This technique can be used to investigate whether individual clusters overlap completely or partially with certain regions of the other modality. An example of such an application is the use of a light microscope image or an HE stained image. With such modalities, the clusters can be examined for overlaps with morphological structures or tissue anomalies, such as disease-related tissue changes.
The highlight layer is not affected by the transparency slider and remains active. Examples for an overlay of a segmentation map and a brightfield image, with and without highlighting, are shown in Figure 5.18E and Figure 5.18F.

***m/z*-image overlay** The m/z-image overlay has several different functions. It can be used to inspect individual m/z-images or groups of m/z-images. If more than one m/z-image is selected, the group is aggregated into a single image. The default aggregation function is the arithmetic mean. Since different intensity distributions can benefit from different color maps, several alternatives are offered. If an m/z-image (or a group) with an intensity distribution of interest is selected, two interesting questions often arise. First, how are the clusters distributed in context with the selected intensity distribution, and second, does one of the clusters overlap with

the selected intensity distribution. To gain insight into these two questions, each cluster can be highlighted or "inversely highlighted" while the m/z-image layer is active. The normal highlighting function works as described above for the segmentation map, i.e. all pixels belonging to the selected cluster are colored white (see Figure 5.18G). The inverse highlighting places a "haze" over every pixel that does not belong to the selected cluster. This is done by drawing over these pixels with white color and a medium transparency value. Using this method, the intensities of all pixels remain visible. As a result, the overlap between the selected cluster and the intensity distribution can be viewed in the context of the overall intensity distribution of the selected m/z-image(s). An example is shown in Figure 5.18H.

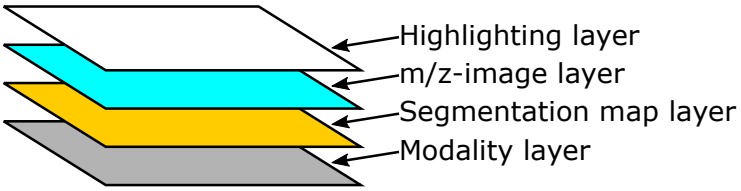


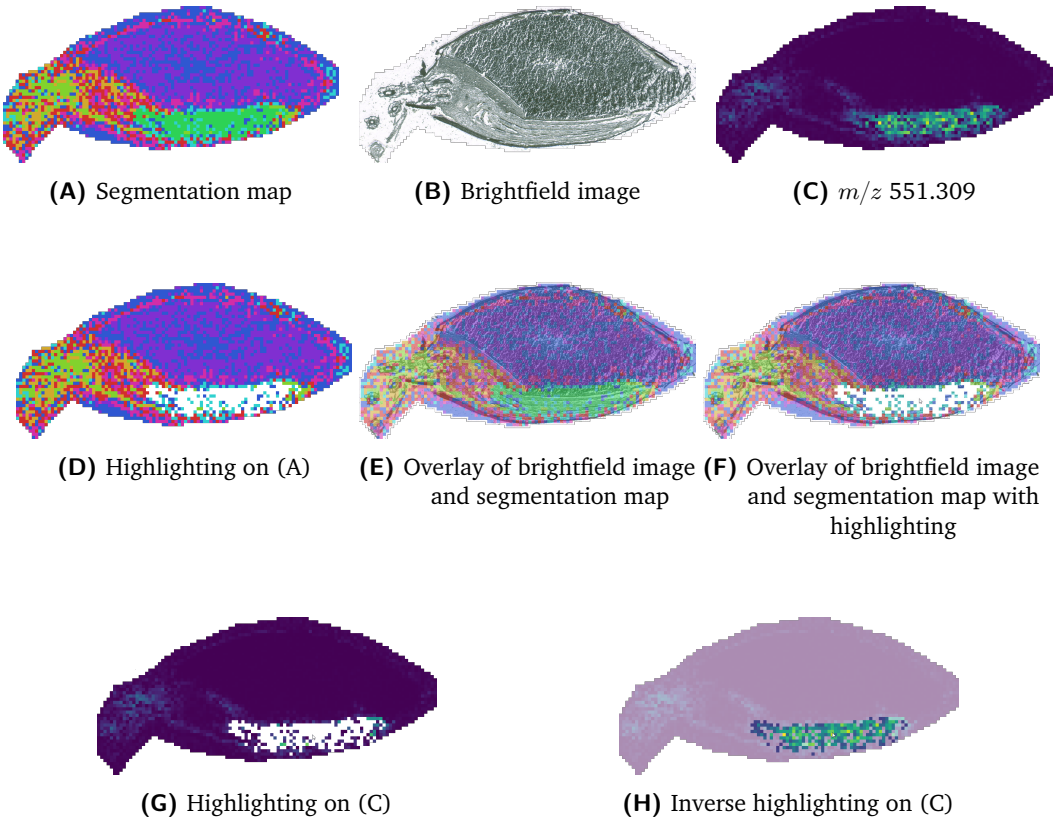
Fig. 5.17.: Layer system of the segmentation map display – Illustration of the layer system used for the segmentation map display.

***m/z*-favorites** The m/z-image overlay functionality is a versatile tool. Not only can it be used to analyze the co-localization of a cluster with intensity distributions of selected m/z-images, but it can also simply act as an m/z-image viewer. With increasing insights in the course of the analysis, it may become necessary to switch frequently between the m/z-images of a subset of m/z-values that have been narrowed down during the analysis. To facilitate the selection, the list display has a "de-favorite" option. By default, all m/z-values in the list display are ordered in descending numerical order. If a subset of m/z-values of interest is distributed over a wide range of values, a frequently alternating selection leads to tedious scrolling and cursor movement. To increase the convenience, m/z-values that are not of particular interest can be de-favorited. After using the de-favorite function, the list display is divided into two different sets:

1. The set of all favorite m/z-values.

2. The set of all non-favorite m/z-values.

Accordingly, the order of the list display is adjusted. The set of favorite m/z-values appears first, still in descending numerical order, followed by the set of non-favorite m/z-values. This reduces the amount of scrolling and cursor movement. In addition, all non-favorite m/z-values are greyed out to distinguish both sets



(A) Segmentation map **(B)** Brightfield image **(C)** m/z 551.309

(D) Highlighting on (A) **(E)** Overlay of brightfield image and segmentation map **(F)** Overlay of brightfield image and segmentation map with highlighting

(G) Highlighting on (C) **(H)** Inverse highlighting on (C)

Fig. 5.18.: Overlay examples – (A) to (C) represent the three individual layers of the segmentation map display using the barley seed data set $\mathcal{I}^{\mathrm{B}}$. The transparency value for (D) and (E) was set to 0.55. The inverse highlight function (G) has a fixed transparency value of 0.55.

visually. Furthermore, this has the additional benefit that presenting results to other people immediately directs the focus to the desired set of m/z-values.

WHIDE for other clustering methods

WHIDE was originally developed for the interactive visual analysis of H^{2}SOM results. However, by using the general clustering procedure together with the adapted projection method as discussed in Section 5.3.3, WHIDE becomes fully compatible with pretty much every clustering method in Euclidean space. Consequently, the full range of interactive exploration possibilities offered by WHIDE becomes available for a variety of clustering methods. This is a major advantage over the application of arbitrary categorical colormaps on static visualizations of segmentation maps.

The prototype coefficient values that are required to produce the CIPRA glyphs are only directly available in prototype-based clustering methods and in combination

with the “cluster-first” variant. Only for this combination, a prototype will be fitted into the original sample space by the clustering method. In order to make WHIDE compatible with both variants presented in Section 5.3.3 (“embed-first” and “cluster-first”) and with prototype-based and non-prototype-based clustering methods, the coefficient values are approximated with a single method that is identical for all approaches.

The coefficient values of a prototype u_r for a particular cluster r are computed as the weighted arithmetic mean of all mass spectra $\rho_z^{(r)}$ that belong to the particular cluster (see Equation (5.5)). The weights $\mathfrak{v}_r$ are computed as the Euclidean distance of the mass spectrum $\rho_z^{(r)}$ to its prototype in the embedded two-dimensional space $\hat{u}_r$.

$$\begin{aligned} \mathfrak{v}_r &= \left\| \rho_z^{(r)}, \hat{u}_r \right\|_2 \\ u_r &= \frac{1}{\sum_r \mathfrak{v}_r} \sum_r \mathfrak{v}_r \rho_z^{(r)} \end{aligned} \tag{5.5}$$

For the projection of the entire embedding, where each pixel is its own prototype, the coefficients are represented by the mass spectrum of the respective pixel.

5.5 Summary and Contributions

In this chapter, we introduced a new and fully revisited version of WHIDE, a tool for the original purpose to allow the interactive visual analysis of H^{2}SOM segmentation maps.
We also extended the projection method, that projects an H^{2}SOM grid structure onto a unit circle, by a position optimization procedure. After position optimization, the coloration better reflects the similarity between clusters.
Next, we combined a general clustering procedure with an adjusted projection method to make a large variety of different clustering methods available for WHIDE. We demonstrated the competitiveness of different variants of this new approach with the H^{2}SOM result by various comparisons.

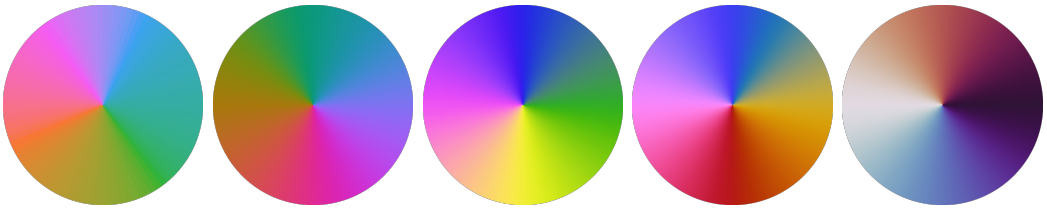
5.6 Improvements and Future Research

Some of the presented results suggested that the visual quality of a segmentation map is likely to be negatively affected if the number of clusters overestimates the intrinsic

dimensionality of the data. This is also indicated by the results of the position optimization. For some pairs of prototypes, the very first steps already led to almost the same positions. This shows that these prototypes are very similar. An interesting follow-up project of the position optimization would be the implementation of a cluster-merge procedure. Similar to position optimization, this method could analyze the similarity of prototypes or cluster compositions and merge them if they are similar enough. We are confident that such a procedure, together with the position optimization, has the potential to lead to an increased visual quality of segmentation maps. Furthermore, this could lead to more homogeneous clusters, which would simplify the analysis with WHIDE.

Another future project could be to investigate how a dendrogram could be projected onto the unit circle. This way, hierarchical clustering might be exploitable in a similar way as the H^2SOM. However, such a mapping would likely require another circular colormap, since in a dendrogram the distance between adjacent clusters can considerably increase in lower hierarchies. Therefore, the HSL colormap is most likely not suitable.

Speaking of colormaps, the analysis of additional circular colormaps is another interesting topic. There is a variety of different circular colormaps with different benefits for different situations [56]. A comprehensive evaluation of such maps for different data sets and analysis objectives could improve the effectiveness and maybe even the efficiency of visual analysis with WHIDE. Some examples of additional colormaps are shown in Figure 5.19.



(A) Seaborn HUSL **(B)** Matplotlib cmocean phase **(C)** Colorcet cyclic mygbm **(D)** Colorcet cyclic mrybm **(E)** Matplotlib twilight

Fig. 5.19.: Examples for cyclic colormaps – Examples of additional cyclic colormaps that could be investigated in WHIDE.

6 Combining the Spatial and Spectral Domain for an Interactive and Responsive Analysis

Parts of this Chapter are based on:
Fast and Visual Exploration of Mass Spectrometry Images with Interactive Dynamic Spectral Similarity Pseudocoloring
`Code: https://github.com/BiodataMiningGroup/quimbi/`

> *Science isn't about why, it's about why not.*
>
> **— Cave Johnson**
> (Founder of Aperture Science)

6.1 Motivation

The previous chapters, Chapter 4 and Chapter 5, discussed analysis approaches on the spatial and spectral domain of MSI data in terms of co-localization and similarity in molecular composition. Both chapters treated these approaches mostly separated. A minor exception is the segmentation maps that were used to present the results of the analysis for similarity in molecular composition.

It has already been shown that the combination of both domains can be beneficial for the analysis of MSI data. An impression of the potential benefits for the analysis of MSI data has already been presented in Alexandrov et al., 2013, where the result of cluster analysis on m/z-images is supplemented by the projection of the cluster assignments to the arithmetic mean spectrum. Although this work shows only a combination of static visualizations that loosely connect both domains, it already illustrates the potential for such combined approaches.

This chapter presents an example that illustrates how a combination of the spatial and spectral domain can improve the analysis of MSI data, by leading to new views on the data and achieving new results.

6.2 Interactive Visual Exploration of Molecular Composition Similarity through Spatial Browsing and Pseudocoloring

To enable interactive analysis of regions with a similar molecular composition that combines and utilizes elements of the spatial and spectral domain of MSI data, we have developed a visual analysis (web)tool called QUIMBI (**qui**ck exploration tool for **m**ultivariate **b**io**i**mages). The basic idea of QUIMBI is to use the spatial domain to interactively explore and evaluate similarities between molecular compositions of mass spectra. The fundamental search and filter interactions to explore the similarities between the mass spectra are performed within the spatial domain. The actual similarity analysis in turn is performed within the spectral domain by computing the angular distance between a selected reference spectrum and all other spectra. The result is then projected back into the spatial domain, which allows an evaluation in relation to the morphology of the sample. The entire tool is interactive and responsive, which means that the similarity computation and projection of the result is performed in real-time during exploration.

The first version of QUIMBI was developed by my colleague Martin Zurowietz in cooperation with Jan Kölling [55]. In this thesis, a fully revised version of QUIMBI is presented, which includes a redesign and a re-implementation. The implementation of this new version started as part of a bachelor project, which was supervised by Martin Zurowietz. Afterwards, the tool was finished in a master project by Christian Fortmann. Most of the implementation work of the revised QUIMBI version was done by both students. A library for running parallel computing with WebGL in QUIMBI was written by Martin Zurowietz for the original version of QUIMBI and adopted for the revised version. Design, conception and supervision of the revised QUIMBI version were done by Martin Zurowietz and myself in cooperation.

Interface

Similar to the other visual analysis tools presented before, the interface of QUIMBI is kept very simplistic. Most parts of the user interface are shown in Figure 6.1. The interface consists of two main visual building blocks:

1. **The spatial-map display**, which shows either a pseudocoloring based on similarity values between the mass spectrum of a selected reference pixel and all other pixel's mass spectra, or the m/z-image of a selected m/z-value.
2. **The spectrum display**, which shows either the mass spectrum of the selected reference pixel or, if no reference pixel is selected, the average mass spectrum of all pixels. To be compatible with picked and unpicked data, the mass spectrum is presented in centroid mode.

Further visual elements are:

- A sidebar to manage all selected spatial and spectral regions of interest (Figure 6.1c). Details are explained below.
- A position indicator (Figure 6.1d) to support orientation, as well as the retrieval and communication of specific reference points.
- A colormap legend (Figure 6.1e) to support the evaluation of the intensity values. The legend is enhanced by a histogram plot to provide an impression of the similarity value distribution.
- Some control buttons to support navigation and selection operators.

Similarity exploration

The interactive exploration of similarities between mass spectra (local molecular compositions) is the main application area of QUIMBI. Due to the high dimensionality of MSI data, it is impossible to present the entirety of all the similarity information within the context of the sample morphology clearly and concisely in a single visualization. QUIMBI approaches this problem with a dynamic and interactive filtering approach. By selecting a reference pixel in the spatial-map display, the pairwise similarity between the mass spectrum of the selected reference pixel ($\mathcal{I}_{h',w'}$) and the mass spectra of all other pixels ($\mathcal{I}_{h,w},\ \forall\ (h,w) \in \rho$) is computed in parallel by the inverse angle between the mass spectra (see Equation (6.4)). The individual similarity values of each mass spectrum are converted into a color $\eta_{h,w}$ using the "Fire" colormap $\mathcal{M}$ (see Equations (6.2) to (6.4)). Subsequently, using the positions

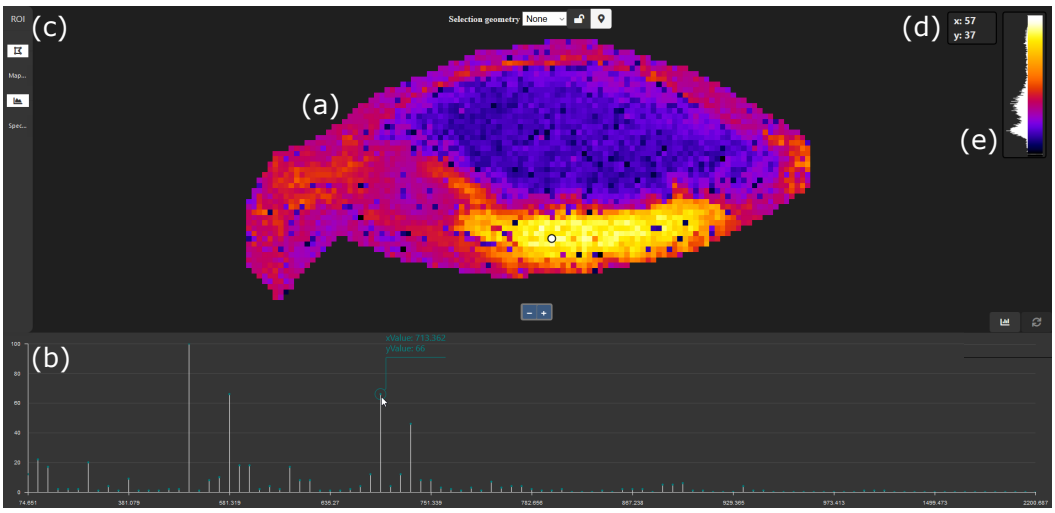


Fig. 6.1.: Overview of the QUIMBI user interface – (a) The spatial-map display, showing the pseudocoloring based on similarity values encoded with the "Fire" colormap. The selected reference pixel is shown as a white circle; (b) The spectrum display, showing the mass spectrum of the selected reference pixel in centroid mode, with an active annotation label that shows information about the m/z-value and the intensity; (c) A collapsed sidebar to manage selected regions of interests; (d) The position indicator; (e) The colormap legend, enhanced by a histogram plot.

h, w associated with each mass spectrum, the similarity values are projected into the spatial domain by coloring each pixel with its respective color within the spatial-map display (exemplified in Figure 6.1). Technical details are provided in Section 6.2.1. In addition, each selection of a reference pixel shows the corresponding mass spectrum in the spectrum display. This allows identifying the molecular composition, i.e. the set of m/z-values for which the similarities are computed. Whenever a new reference pixel is selected, the colors of all pixels are updated dynamically according to the new similarity values. Also, the displayed mass spectrum is adjusted.

The pseudocolored distribution pattern represents the quantified similarity of the molecular composition from a selected position to all other local molecular compositions across the entire tissue. Consequently, it describes which spatial regions of the tissue have a similar molecular composition to the selected position. For this reason, these distribution images will be referred to as mcs-images (molecular composition similarity images) in the following.

The smoothness of the interactive exploration and the dynamically changing colors of the mcs-images allow an immediate perception of the relationship between the molecular compositions of mass spectra and the morphology of the sample [73]. Without any prior knowledge, the exploration of the different mcs-images can support the identification of molecular similar regions. However, if interesting spatial

locations are known, e.g. due to prior histopathological staining, these locations can be specifically selected. This allows to search for regions within the sample that exhibit a similar molecular composition to the mass spectrum of the reference location, but may not be recognizable by the histopathological staining.

In principle, an mcs-image looks like a pseudocolored m/z-image and can be interpreted as a kind of soft-segmentation map. This facilitates accessibility and interpretation and thus reduces the learning curve of QUIMBI. Due to the implementation with parallel computing, a simple cursor hover interaction is sufficient to perform a selection. However, for an in-depth analysis of the similarity distribution pattern, as well as the analysis of the reference spectrum with the annotation labels, it is necessary to be able to lock the current selection. Therefore, a selected reference pixel can also be locked, which is visually emphasized by a white circle. A lock disables the responsiveness of the spatial-map display until it is resolved.

Finally, zoom and drag interactions are available to support the analysis of the mcs-images at different levels of detail.

Regions of Interest Selection

A region of interest (RoI) is a selected subset of data points from a given data set. This selection can be done manually or computationally. For data analysis, especially high-dimensional and high-volume data, there are numerous reasons for the selection and analysis of individual RoIs. The selection of RoIs presents a powerful filter technique that allows reducing the amount of data by removing question- or task-irrelevant parts either temporarily or permanently. This offers several benefits, such as: the computational resource requirements are reduced; new or different insights, structures and details can be uncovered; question- and task-irrelevant parts of the data can be removed, which otherwise can negatively influence the analysis and may have an irritating effect that negatively influences the focus of the user, which is especially true for visual analysis. RoIs are even more powerful when used as a temporary selection. This way the context of a RoI can be added or removed dynamically.

To make use of this powerful filter technique, QUIMBI allows defining spatial and spectral RoIs. Since a mass spectrum is one-dimensional, the spectral RoI selection is defined as a continuous range of m/z-values. However, separated value ranges can be combined by activating multiple spectral RoIs at a time. Unlike the spectral domain, the spatial domain is two-dimensional. Therefore, a RoI is defined as an

arbitrary polygon. As before, separated RoIs can be combined by activating multiple spatial RoIs at a time.

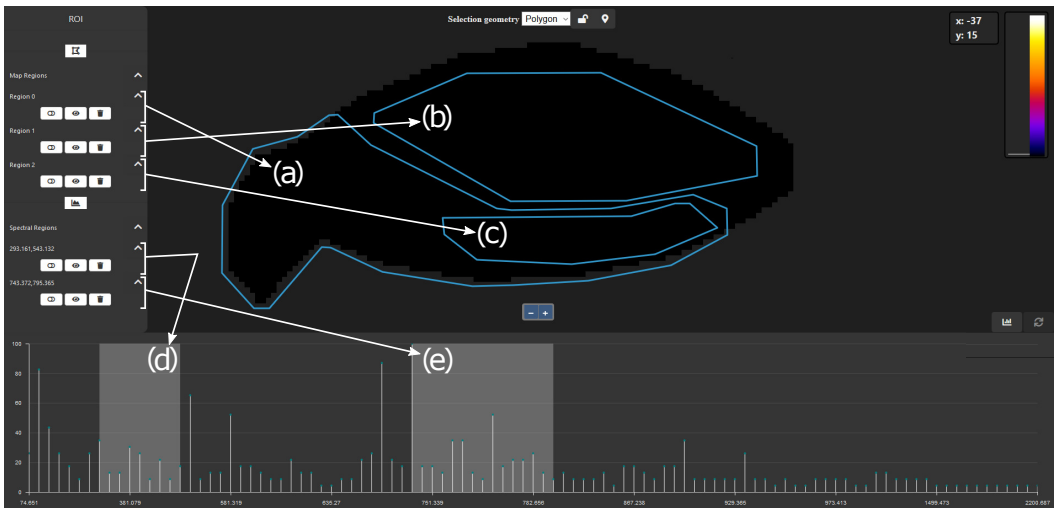
In order to enable a dynamic activation and deactivation, each RoI can be managed individually in QUIMBI. This means that both the visual outlines and the functional selection of each RoI can be activated and deactivated individually. Consequently, all RoIs can be combined freely. This also applies to combinations of spatial and spectral RoIs. Examples for different RoIs and the management sidebar are shown in Figure 6.2A.

A functional activation restricts all computations in the corresponding domain to the combination of all active RoIs. Examples for spatial and spectral RoIs, as well as an illustration of how they can affect the analysis and results are shown in Figure 6.2. The activation of the visual outlines is for control purposes only and does not affect the computations.

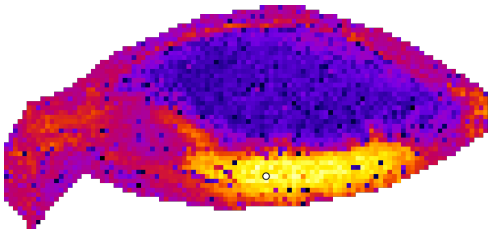
Spectrum browsing

Several of the operations available in QUIMBI can benefit from the information provided by the molecular distributions of the individual m/z-images. Therefore, the spectrum display can be used as an m/z-image browser. During spectral browsing, the color values $\eta_{h,w}$ of each pixel in the spatial-map display are computed based on the intensity values of the m/z-images (see Equations (6.2), (6.3) and (6.5)). The access to m/z-images helps to obtain an overview and an overall understanding of the data set, which supports the evaluation of mcs-images. It also allows the identification of m/z-values with interesting distribution patterns, which in turn can support the selection of spatial RoIs.

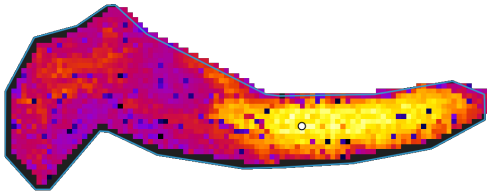
Spectral RoIs can be used to visualize the averaged m/z-image of all m/z-values selected through the combination of all functionally activated spectral RoIs $Z' \subseteq \{0, \ldots, Z-1\}$. In this case, the color values $\eta_{h,w}$ of each pixel in the spatial-map display are computed according to Equations (6.2), (6.3) and (6.6). Furthermore, spectral browsing is also possible with functionally activated spatial RoIs, but this can influence the exact color assignment of individual pixels due to the normalization step in Equation (6.3). During spectral browsing, the displayed spectrum is restricted to the arithmetic mean of all mass spectra. This avoids interference problems with the spatial-map display and other operations. To keep this function easy and intuitive, browsing is implemented as a simple hover interaction. While browsing the spectrum, an annotation label provides information about the m/z-value and the intensity.



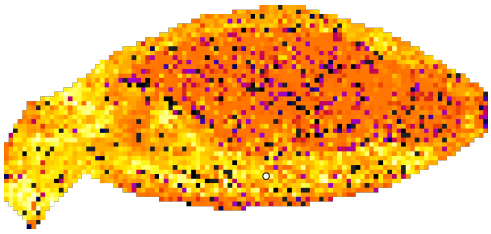
(A) QUIMBI interface with an expanded region of interest (RoI) sidebar and multiple selected RoIs.



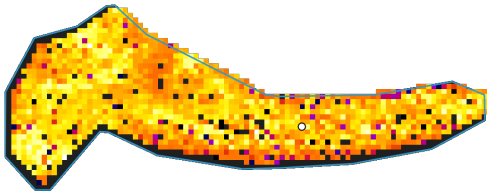
(B) No RoI is active.



(C) Spatial RoI (a) active.



(D) Spectral RoI (e) active.



(E) Spatial RoI (a) and spectral RoI (e) active.

Fig. 6.2.: Region of interest examples – (A) Overview of the QUIMBI interface with no reference pixel selected. Three spatial RoIs (a,b,c) and two selected spectral RoIs (d,e) are selected. All RoIs are visible but functionally deactivated. (B) to (E) show mcs-images for a selected reference pixel ($x = 57, y = 37$) and different functionally activated RoIs.

If a set of spatial RoIs is activated, the mean mass spectrum presented in the spectrum display can be recomputed. This restricts the computation to the spectra of the activated RoIs, which in turn allows gaining an impression of the characteristic molecular composition of the activated RoIs.

To avoid interference with the dynamic response of the spatial-map display, selecting a reference pixel blocks the spectral browsing functionality. This way, the reference mass spectrum can be examined with the annotation labels without losing the context of the mcs-image. Similarly, the responsiveness of the spatial-map display can be blocked to define spatial regions of interest with the context of an m/z-image.

As for the spatial-map display, zoom and drag interactions are available to support the analysis of the mass spectra at various levels of detail.

6.2.1 Efficient Implementation of QUIMBI

QUIMBI uses the graphics processing unit (GPU) for hardware-accelerated computation of the interactive visualization in real-time. Since QUIMBI is a web application, the GPU is exposed through the WebGL application programming interface (API). WebGL is intended to be used to display two- or three-dimensional graphics in a web browser. In order to process much higher dimensional data, QUIMBI uses the WebGL API in a special way, which is outlined in the following.

A typical graphics rendering pipeline with WebGL consists of four steps:

Step 1: GPU memory allocation and initialization of the WebGL shader programs which contain the logic that should be executed in the GPU. The GPU memory is typically used to store textures, i.e. images that are projected onto two- or three-dimensional surfaces when the rendering pipeline is executed. The total size of the available texture memory varies depending on the model of the GPU. However, the texture memory layout is always a two-dimensional array of 4-byte values, one each for the red, green, blue and alpha color channels of an image. To fit an MSI data set $\mathcal{I}$, which consists of Z m/z-images, into the texture memory, the individual intensity values $\mathcal{I}_{h,w,z}$ are first transformed into bytes $\widetilde{\mathcal{I}}_{h,w,z} \in \{x \in \mathbb{N} \mid 0 \leqslant x \leqslant 255\}$ (see Equation (6.1)). The result is a 8-bit normalized data set $\widetilde{\mathcal{I}}$. The normalized data set $\widetilde{\mathcal{I}}$ is split into tiles, where each tile consists of four consecutive m/z-images as illustrated in Figure 6.3. Each tile can be treated like a regular image, except that the four color channels of the regular image now correspond to four m/z-images of the data set. All tiles of a data set can be loaded into the texture memory of the GPU and are available to be accessed by the WebGL shader programs.

$$\widetilde{\mathcal{I}}_{i,j,z} = \left[255 \cdot \frac{\mathcal{I}_{i,j,z}}{\max_{i,j,z}(\mathcal{I}_{i,j,z})}\right], \text{where } [\cdot] \text{ denotes the rounding operation} \quad (6.1)$$

Step 2: A vertex shader program computes a scene of two- or three-dimensional shapes. In the case of QUIMBI, this scene consists only of a single two-dimensional rectangle onto which the image of the data visualization (mcs-image or m/z-image) is to be projected for display.

Step 3: A fragment shader program computes the color of each pixel of the scene to be rendered. The fragment shader program is executed for each pixel in parallel on the GPU, which speeds up the computation considerably. The color of a pixel is typically based on a single interpolated color from a texture, as well as computed shadows or reflections in a three-dimensional scene. However, to compute the color of a pixel for the visualization in QUIMBI, the fragment shader program needs to access the colors from the appropriate positions of all the image tiles stored in the texture memory. To compute the color $\eta_{h,w}$ of a pixel in the visualization, different fragment shader programs are executed, depending on the visualization mode (mcs-image (Equation (6.4)), m/z-image (Equation (6.5)) or mean m/z-image (Equation (6.6))). Each fragment shader program first computes a similarity/intensity value $\eta'_{h,w}$. This is followed by the adaptive colormap optimization to compute $\eta''_{h,w}$ (see Equation (6.3)) [28], which is then transformed to the final color $\eta_{h,w}$ of the pixel using the colormap $\mathcal{M}$ (see Equation (6.2)).

Step 4: In this final step of the WebGL rendering pipeline, the image generated by the fragment shader program is returned from GPU memory and drawn in the spatial-map display. For the interactive real-time visualization, the entire WebGL rendering pipeline is executed in a loop. The visualization is updated whenever either the cursor position hovers a pixel in the spatial-map display, the m/z-value changes during spectral browsing, or a spectral RoI is activated.

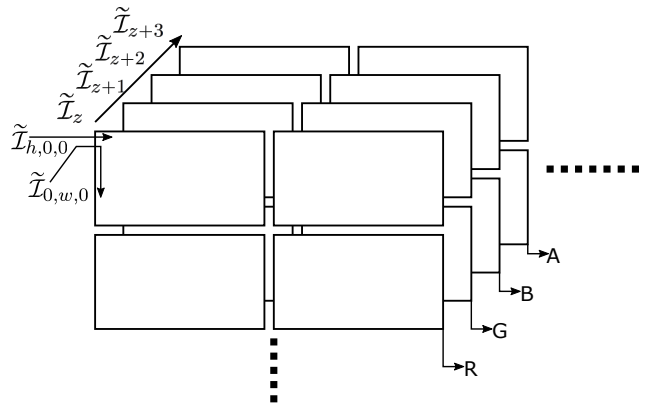


Fig. 6.3.: Tile layout – Illustration of the tile layout in texture memory to store the byte transformed MSI data set $\widetilde{\mathcal{I}}$.

$$\eta_{h,w} = \mathcal{M}(\eta''_{h,w}) \tag{6.2}$$

$$\eta''_{h,w} = 255 \frac{\eta'_{h,w} - \min_{h,w} \eta'_{h,w}}{\max_{h,w} \eta'_{h,w} - \min_{h,w} \eta'_{h,w}} \tag{6.3}$$

$$\eta'_{h,w} = \left[255 \left(1 - \frac{2}{\pi} \arccos \left(\frac{\widetilde{\mathcal{I}}_{h,w} \cdot \widetilde{\mathcal{I}}_{h',w'}}{|\widetilde{\mathcal{I}}_{h,w}||\widetilde{\mathcal{I}}_{h',w'}|} \right) \right) \right] \tag{6.4}$$

$$\eta'_{h,w} = \widetilde{\mathcal{I}}_{h,w,z} \tag{6.5}$$

$$\eta'_{h,w} = \left[255 \frac{1}{|Z'|} \sum_{z \in Z'} \widetilde{\mathcal{I}}_{h,w,z} \right] \tag{6.6}$$

where $\cdot$ denotes the dot product and $[\cdot]$ denotes the rounding operator.

6.2.2 Advantages and Limitations

Advantages of parallelization QUIMBI scales well with the spatial resolution due to the implementation in WebGL, which features a parallel computation of Equation (6.4) using the GPU. This allows the computation of similarities in real-time and makes the visualization with QUIMBI very responsive.

Limitations Due to the WebGL implementation, the limiting computational resources are GPU speed, GPU memory and CPU memory. Apart from that, the use of WebGL brings some additional limitations. As explained in Section 6.2.1, the data set $\mathcal{I}$ needs to be transformed into $\widetilde{\mathcal{I}}$, which requires a transformation of every intensity value into 8-bit. Therefore the theoretical maximal resolution of different intensity values per mass spectrum is limited to only 256 values. Another limitation is the spectral resolution, i.e. the number of total m/z-values. The higher the spectral resolution, the more computation time per GPU core is required to compute the similarity values. The similarity computation for a single pixel pair is not parallelized, i.e. it does not scale with the number of cores, but with the power of each individual core.

6.2.3 Application on Real Data

QUIMBIs main area of application is the interactive exploration of mcs-images. To illustrate the benefits of the visual analysis with QUIMBI, two different applications are demonstrated (see Figure 6.4 and Figure 6.5).

Remark For the following application examples, the human skin $\mathcal{I}^{\mathrm{S}}$ and the mouse kidney $\mathcal{I}^{\mathrm{K}}$ data sets were each used without any square-root or logarithm transformation. This is because the reduced variance of peak intensities caused by the transformation produced a poorer color contrast in QUIMBI for these data sets. Furthermore, $\mathcal{I}^{\mathrm{S}}$ was re-picked with a threshold of 0.0052 (198 peaks) because the original number of 51 peaks was too low to reveal the distinct morphological regions with similar molecular composition.

Figure 6.4 shows different mcs-images for the barley seed $\mathcal{I}^{\mathrm{B}}$, the mouse kidney $\mathcal{I}^{\mathrm{K}}$ and the mouse urinary bladder $\mathcal{I}^{\mathrm{U}}$ data sets. Each of these mcs-images emphasizes a different known morphological structure of the respective sample. This demonstrates that QUIMBI can be used to explore morphological structures that feature a similar molecular composition. While similar analysis results might be achieved by a tedious analysis of many different individual m/z-image distributions or static segmentation maps, the analysis with QUIMBI is much faster and offers more flexibility through its responsiveness.

Figure 6.5 uses the human skin data set $\mathcal{I}^{\mathrm{S}}$ to demonstrate how QUIMBI can be used to reveal morphological regions with similar molecular compositions that cannot be identified by the analysis of individual m/z-images like in the example before. The presented mcs-image in Figure 6.5 reveals a weak but distinct distribution pattern of pixels that have a similar molecular composition to the reference pixel (white arrow). The mass spectrum of the selected reference pixel is presented in Figure 6.5C. The distribution pattern shown can neither be found by inspecting all of the individual m/z-images (Figures 6.5D to 6.5P), nor by inspecting their arithmetic mean. This example shows that QUIMBI is not only useful for accelerating the traditional single m/z-image analysis but can also provide new insights that the analysis of individual m/z-images cannot provide.

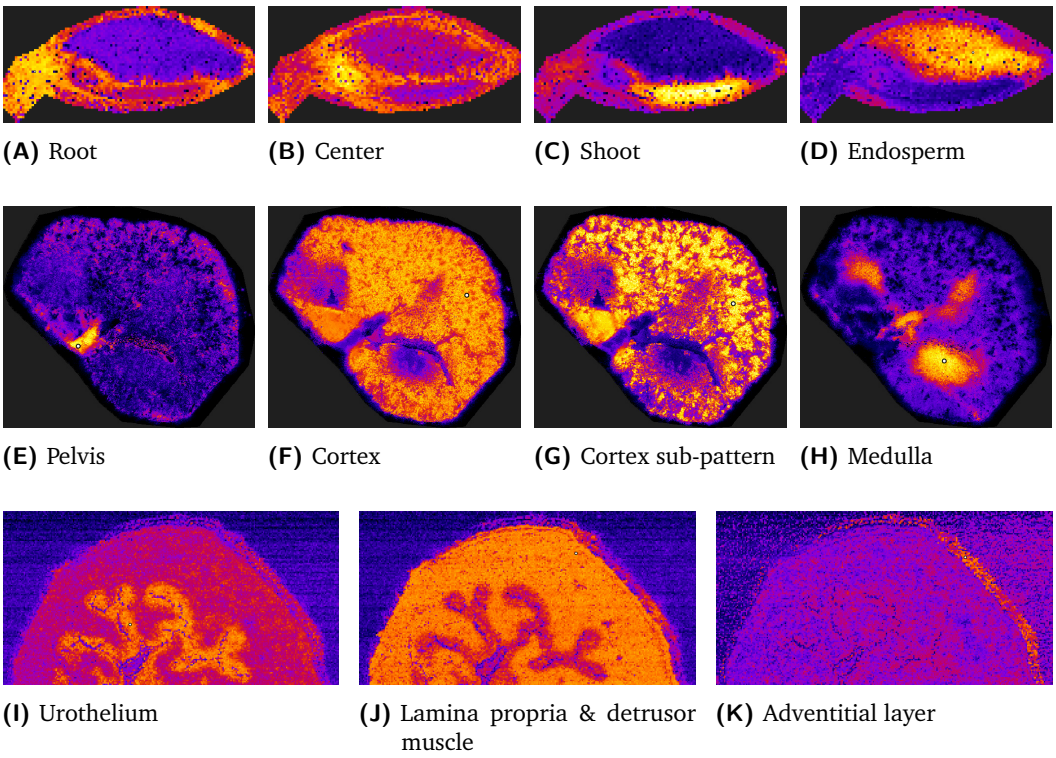
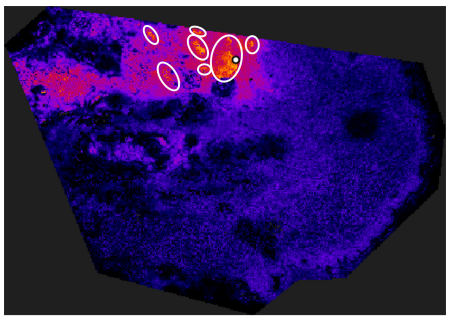
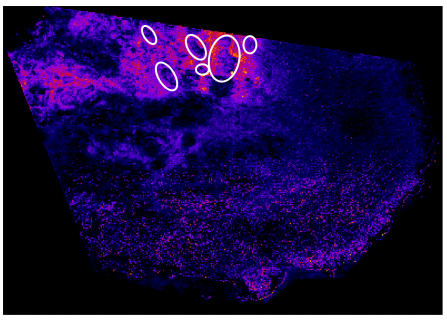


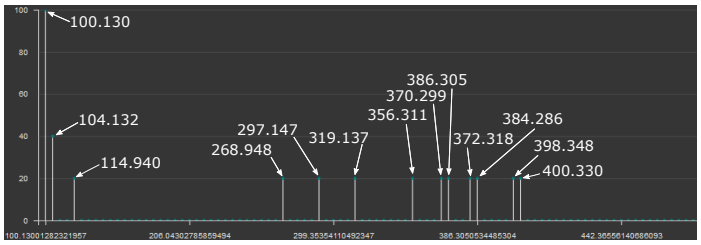
Fig. 6.4.: QUIMBI visualizations – Examples of mcs-images for the barley seed ((A) to (D)), mouse kidney ((E) to (H)) and mouse urinary bladder ((I) to (K)) data sets. Each mcs-image shows regions with a similar molecular composition that corresponds to known morphological regions.



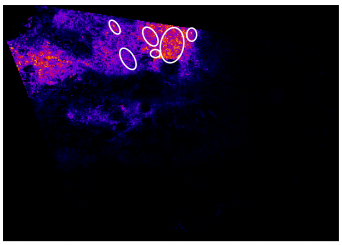
(A) Mcs-image



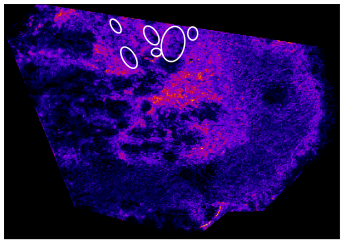
(B) Mean m/z-image



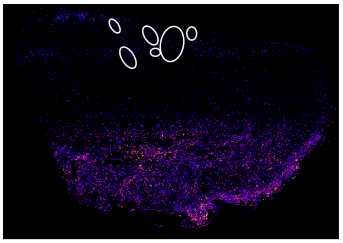
(C) Spectrum of selected pixel in (A)



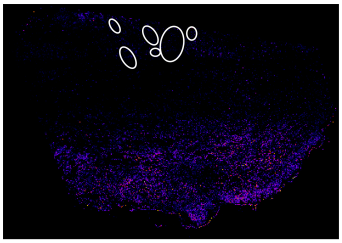
(D) m/z 100.130



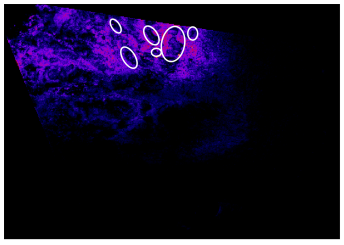
(E) m/z 104.132



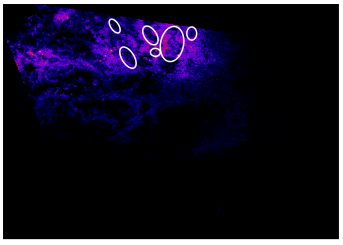
(F) m/z 114.940



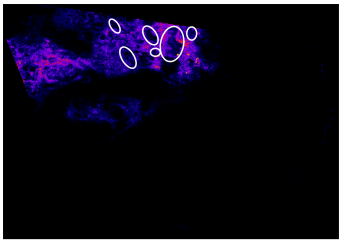
(G) m/z 268.948



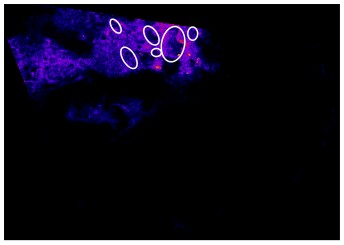
(H) m/z 297.147



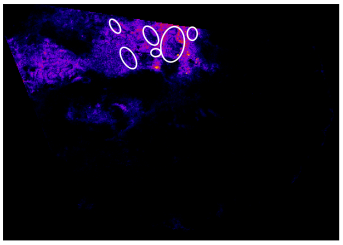
(I) m/z 319.137



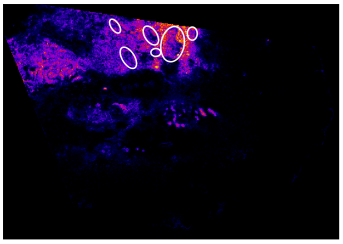
(J) m/z 356.311



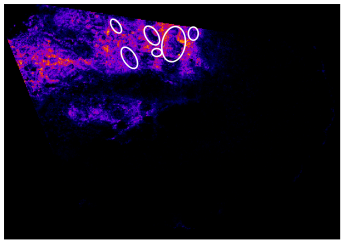
(K) m/z 370.299



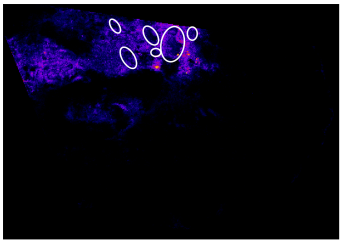
(L) m/z 372.318



(M) m/z 384.286



(N) m/z 386.305



(O) m/z 398.348

(P) m/z 400.330

Fig. 6.5.: Revelation of an mcs-unique distribution – Example of a human skin mcs-image (A) that shows a distribution of molecularly similar positions that cannot be found in the individual m/z-images (D) to (P) or their arithmetic mean (B).

6.3 Summary and Contributions

In this chapter, we presented a revised version of QUIMBI, a visual analysis tool for the interactive exploration of morphological regions with similar molecular composition. QUIMBI enables this analysis by allowing the user to browse through the spatial dimension of the sample to examine the distribution of similarity relationships between mass spectra.

We briefly discussed the advantages and disadvantages of parallelization in QUIMBI and showed two application examples. The first example demonstrated how QUIMBI can be used to accelerate the traditional analysis of single m/z-images and their connection to the sample morphology, i.e. the examination and identification of layers and structures in a tissue section. The second example demonstrated that due to the interactive exploration on the spatial dimension (the morphology of the sample), instead of the mass spectrum, interesting distribution patterns become visible, which would not be found when examining only single or multiple m/z-images. This means that new and previously unknown morphological regions with similar molecular compositions can be uncovered.

In summary, QUIMBI offers an attractive way to discover interesting structures and hidden regularities in tissue sections. This information gain can help to understand the (molecular) nature of (unknown) regions and to accelerate the process of identifying regions with similar molecular compositions, which could also be interesting for fields like research pathology or clinical pathology.

6.4 Improvements and Future Research

QUIMBI still has a few missing functions, whose addition could improve the potential of the tool considerably. Examples are: the addition of more colormaps, such as "viridis", "magma" or "inferno"; the ability to mark single m/z-values of interest and their annotation to make the spectrum browsing function more targeted; the ability to add another imaging modality as background (e.g. a histopathologically stained image or a microscopy image) and to provide overlay view between the mcs-images and this imaging modality.

7 Conclusion

> *It always seems impossible until it's done.*
>
> **— Nelson Mandela**
> (Activist and politician)

In this thesis, we have explored the importance of visual semi-automated analysis for the field of mass spectrometry imaging. For this purpose, we examined the spatial domain and the spectral domain of MSI data sets, individually and in combination. We applied various data mining and information visualization techniques and showed how these techniques can vastly increase the value of MSI data.

To ensure compatibility with all of the visual analysis tools developed, we presented a (pre-)processing pipeline (ProViM) that covers the usual processing steps of signal alignment, normalization and peak picking. In addition, to these commonly used steps, we presented an interactive dimension reduction-based method that allows to detect matrix and artifact signals and to reduce their influence on the data set by subtracting their profiles in form of mean spectra. We showed that some MSI data sets do not only feature matrix and sample characterizing spectra, but also spectra that do not fit either the matrix or the sample, which we classified as artifacts. We could also demonstrate that the subtraction of matrix signals can enhance the signal intensity of a variety of sample-specific signals, especially for larger metabolites like lipids in the higher Dalton mass range. Furthermore, we presented a pattern-based approximation algorithm for the detection and removal of isotope signals to reduce the amount of duplicated information after peak picking, i.e. m/z-values that describe the same molecule and consequently feature the same m/z-image.

In order to allow a fast initial assessment and comparison of raw and processed data, we developed a visual analysis tool for interactive data exploration based on various dimensional reduction embeddings (VAIDRA). Dimension reduction is a commonly used analysis technique in the field of MSI. However, not all dimension reduction methods are equally well suited to explore all kinds of MSI data and all kinds of research questions. Another commonly known problem is that the projection of the data on some embedding axes might only resolve trivial separations, e.g. matrix and sample separation in the first embedding axis, which corresponds to a trivial

foreground and background separation. Thus, an interactive visual analysis tool that allows comparing embeddings computed with different dimension reduction methods and that allows to dynamically explore two-dimensional sub-embeddings is of great value. The combination of different m/z-embedding-images into an RGB projection image, which can be interpreted as a kind of "soft-segmentation map", often provides a good early overview of the spatial structure of the data. We also extended the commonly provided interactivity of the scatterplot with interaction possibilities on the m/z-embedding-images, the m/z-images and the RGB projection image. This enables the user to explore the embeddings based on known morphological features or molecular distributions of interest.

To investigate the spatial domain we conducted a comprehensive study of relationships between m/z-images, the use of pre-processing procedures, several similarity functions and different clustering methods. We found that classic and commonly used functions like Pearson correlation and cosine similarity, combined with preprocessing are often suited to analyze the spatial domain of an MSI data set. This is especially valid for samples with regular morphological structure. The results also indicated an added benefit for various samples using different scale-spaces to emphasize structures of different granularity. However, the results also showed that this statement is not fully generalizable. In order to maximize the value of an analysis, the precise setup of an analysis pipeline should be specifically designed and evaluated for any combination of data set and research objective. Hence, we proposed a workflow to evaluate different pipeline setups (SoRC). While SoRC provides a good first step for the automated evaluation of pipeline setups, it still has some severe weaknesses due to inherent problems of the evaluation metrics. To fully exploit the potential of such a procedure an extension is required that allows to address more characteristics of the analysis result.
To investigate the potential of visual analysis for the spatial domain, we proposed to use community detection as a clustering method. For this purpose, we developed a procedure to map the relationship between individual m/z-images, i.e. molecular distributions, on a graph structure. Since this is a non-trivial problem, we presented a set of different methods that worked reliably for the tested data sets. The application on the barley seed data set showed good results for the task of classifying individual metabolite classes. The interactive visual analysis tool (GRINE) allows to track relations intuitively and to investigate and understand cluster decisions, which can be difficult for clustering methods that are not graph-based. GRINE also uses this benefit to allow a manual correction of clustering decisions, which allows to incorporate existing expert knowledge into the clustering result. By using a combined visual representation of graph and adjacency matrix even very complex

relations between molecular distributions can be investigated.
Furthermore, we presented a method to approximate regions of interest for clusters of m/z-images. We presented some promising results for the barley seed data set. However, the method is still experimental and has some major weaknesses that need to be addressed.

To investigate the spectral domain, we focused on segmentation maps. We proposed to use the H^{2}SOM algorithm to compute MSI segmentation maps as this method provides an intuitive option for color-coding, which uses a unit circle projection that maps the H^{2}SOM onto a color disc. By taking advantage of the neighborhood preserving characteristic of the H^{2}SOM we presented an algorithm to improve the color contrast of the computed segmentation maps, which increased the perception of morphological structures. To investigate the potential of visual analysis for the spectral domain, we developed a revised version of a previously presented visual analysis tool for the interactive exploration of segmentation maps (WHIDE). WHIDE benefits the exploration of MSI segmentation maps in various ways. It can be used to interactively adapt the color contrast to emphasize specific areas of interest, it allows a comparison of segmentation maps with individual m/z-images, as well as groups of m/z-images and other imaging modalities, and it provides an easy visualization to identify the most characteristic features (m/z-values) of each cluster. We also presented a projection method that makes the color circle projection applicable to virtually any clustering method in Euclidean space and even any clustering method whose result can be projected into Euclidean space. That way, the application possibilities of WHIDE were greatly increased.

As the visual analysis in both MSI domains, spatial and spectral, showed to be very beneficial for the exploration and analysis of MSI data. The subsequent consideration was to combine the two domains for the interactive visual analysis. We exemplified this approach with a revised version of a previously presented visual analysis tool (QUIMBI). By using the spatial browsing functionality of QUIMBI, we found an example distribution of mass spectra with similar molecular composition that was not identifiable through the analysis of individual m/z-images or their combination by using the arithmetic mean. Based on our experience with QUIMBI, we assume that there is a great potential for further visual analysis tools that combine the spatial and the spectral MSI domain.

In summary, this thesis contributes to the interdisciplinary field of molecular life sciences, computer science and visual analytics by providing various new methods and tools for the semi-automated interactive visual analysis of MSI data. The application of these various methods and tools demonstrated the importance of

semi-automated visual analysis in the field of MSI and the great benefit they can add for the exploration, analysis and interpretation of MSI data.
We hope that the presented methods and tools will help researchers in their analysis and thereby increase the value of their data and that we could contribute our part to give visual analysis even more attention in the field of mass spectrometry imaging.

Bibliography

[1] Ayman Abaza, Mary Ann Harrison, and Thirimachos Bourlai. “Quality metrics for practical face recognition”. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE. 2012, pp. 3103–3107 (cit. on p. 69).

[2] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. “On the surprising behavior of distance metrics in high dimensional space”. In: *International conference on database theory*. Springer. 2001, pp. 420–434 (cit. on p. 35).

[3] Aiman Alam-Nazki and J Krishnan. “Spatial control of biochemical modification cascades and pathways”. In: *Biophysical journal* 108.12 (2015), pp. 2912–2924 (cit. on p. 2).

[4] Theodore Alexandrov, Michael Becker, Sören-oliver Deininger, et al. “Spatial segmentation of imaging mass spectrometry data with edge-preserving image denoising and clustering”. In: *Journal of proteome research* 9.12 (2010), pp. 6535–6546 (cit. on pp. 9, 138, 143).

[5] Theodore Alexandrov, Michael Becker, Orlando Guntinas-Lichius, Günther Ernst, and Ferdinand von Eggeling. “MALDI-imaging segmentation is a powerful tool for spatial functional proteomic analysis of human larynx carcinoma”. In: *Journal of cancer research and clinical oncology* 139.1 (2013), pp. 85–95 (cit. on p. 138).

[6] Theodore Alexandrov, Ilya Chernyavsky, Michael Becker, Ferdinand von Eggeling, and Sergey Nikolenko. “Analysis and interpretation of imaging mass spectrometry data by clustering mass-to-charge images according to their spatial similarity”. In: *Analytical chemistry* 85.23 (2013), pp. 11189–11195 (cit. on pp. 8, 17, 179).

[7] Theodore Alexandrov and Jan Hendrik Kobarg. “Efficient spatial segmentation of large imaging mass spectrometry datasets with spatially aware clustering”. In: *Bioinformatics* 27.13 (2011), pp. i230–i238 (cit. on pp. 9, 138, 143).

[8] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. “OPTICS: ordering points to identify the clustering structure”. In: *ACM Sigmod record* 28.2 (1999), pp. 49–60 (cit. on p. 89).

[9] R Bellman. “Dynamic programming princeton university press princeton”. In: *New Jersey Google Scholar* (1957) (cit. on p. 34).

[10] Richard E Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 1961 (cit. on p. 34).

[11] Kyle D Bemis, April Harry, Livia S Eberlin, et al. “Cardinal: an R package for statistical analysis of mass spectrometry-based imaging experiments”. In: *Bioinformatics* 31.14 (2015), pp. 2418–2420 (cit. on p. 39).

[12]Kyle D Bemis, April Harry, Livia S Eberlin, et al. “Probabilistic segmentation of mass spectrometry (MS) images helps select important ions and characterize confidence in the resulting segments”. In: *Molecular & Cellular Proteomics* 15.5 (2016), pp. 1761–1772 (cit. on pp. 138, 143).

[13]Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. “When is “nearest neighbor” meaningful?” In: *International conference on database theory*. Springer. 1999, pp. 217–235 (cit. on p. 35).

[14]Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. “Fast unfolding of communities in large networks”. In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008 (cit. on pp. 110, 114).

[15]Mark T Bokhart, Milad Nazari, Kenneth P Garrard, and David C Muddiman. “MSiReader v1. 0: evolving open-source mass spectrometry imaging software for targeted and untargeted analyses”. In: *Journal of the American Society for Mass Spectrometry* 29.1 (2018), pp. 8–16 (cit. on p. 39).

[16]Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. “D^3 data-driven documents”. In: *IEEE transactions on visualization and computer graphics* 17.12 (2011), pp. 2301–2309 (cit. on p. 120).

[17]Tadeusz Caliński and Jerzy Harabasz. “A dendrite method for cluster analysis”. In: *Communications in Statistics-theory and Methods* 3.1 (1974), pp. 1–27 (cit. on pp. 78, 110).

[18]Richard M Caprioli, Terry B Farmer, and Jocelyn Gile. “Molecular imaging of biological samples: localization of peptides and proteins using MALDI-TOF MS”. In: *Analytical chemistry* 69.23 (1997), pp. 4751–4760 (cit. on p. 4).

[19]John M Chambers. *Graphical methods for data analysis*. CRC Press, 2018 (cit. on p. 169).

[20]Herman Chernoff. “The use of faces to represent points in k-dimensional space graphically”. In: *Journal of the American statistical Association* 68.342 (1973), pp. 361–368 (cit. on p. 169).

[21]William S Cleveland and William S Cleveland. “A color-caused optical illusion on a statistical graph”. In: *The American Statistician* 37.2 (1983), pp. 101–105 (cit. on pp. 16, 17, 125).

[22]Cynthia Martins Villar Couto, César Henrique Comin, and Luciano da Fontoura Costa. “Effects of threshold on the topology of gene co-expression networks”. In: *Molecular BioSystems* 13.10 (2017), pp. 2024–2035 (cit. on p. 103).

[23]David L Davies and Donald W Bouldin. “A cluster separation measure”. In: *IEEE transactions on pattern analysis and machine intelligence* 2 (1979), pp. 224–227 (cit. on p. 110).

[24]Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. “Indexing by latent semantic analysis”. In: *Journal of the American society for information science* 41.6 (1990), pp. 391–407 (cit. on p. 44).

[25] Soren-Oliver Deininger, Matthias P Ebert, Arne Futterer, Marc Gerhard, and Christoph Rocken. “MALDI imaging combined with hierarchical clustering as a new tool for the interpretation of complex human cancers”. In: *Journal of proteome research* 7.12 (2008), pp. 5230–5236 (cit. on pp. 9, 143).

[26] Alex Dexter, Alan M Race, Rory T Steven, et al. “Two-phase and graph-based clustering methods for accurate and efficient segmentation of large mass spectrometry images”. In: *Analytical chemistry* 89.21 (2017), pp. 11293–11300 (cit. on p. 138).

[27] Zhuanlian Ding, Xingyi Zhang, Dengdi Sun, and Bin Luo. “Overlapping community detection based on network decomposition”. In: *Scientific reports* 6 (2016), p. 24115 (cit. on p. 114).

[28] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. “Color lens: Adaptive color scale optimization for visual exploration”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.6 (2010), pp. 795–807 (cit. on p. 187).

[29] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *Kdd*. Vol. 96. 34. 1996, pp. 226–231 (cit. on p. 89).

[30] Vladimir Estivill-Castro. “Why so many clustering algorithms: a position paper”. In: *ACM SIGKDD explorations newsletter* 4.1 (2002), pp. 65–75 (cit. on p. 8).

[31] Judith M Fonville, Claire Carter, Olivier Cloarec, et al. “Robust data processing and normalization strategy for MALDI mass spectrometric imaging”. In: *Analytical chemistry* 84.3 (2012), pp. 1310–1319 (cit. on pp. 39, 41).

[32] Judith M Fonville, Claire L Carter, Luis Pizarro, et al. “Hyperspectral visualization of mass spectrometry imaging data”. In: *Analytical chemistry* 85.3 (2013), pp. 1415–1423 (cit. on p. 63).

[33] Linton C Freeman. “A set of measures of centrality based on betweenness”. In: *Sociometry* (1977), pp. 35–41 (cit. on p. 132).

[34] Brendan J Frey and Delbert Dueck. “Clustering by passing messages between data points”. In: *science* 315.5814 (2007), pp. 972–976 (cit. on pp. 88, 89).

[35] Akshay Gore and Savita Gupta. “Full reference image quality metrics for JPEG compressed images”. In: *AEU-International Journal of Electronics and Communications* 69.2 (2015), pp. 604–608 (cit. on p. 84).

[36] Karin Gorzolka, Jan Kölling, Tim W Nattkemper, and Karsten Niehaus. “Spatio-temporal metabolite profiling of the barley germination process by MALDI MS imaging”. In: *PLoS One* 11.3 (2016) (cit. on pp. 22, 134, 138).

[37] The HDF Group. *The HDF5® Library & File Format*. `https://www.hdfgroup.org/solutions/hdf5/`. Accessed: 2020-10-26 (cit. on p. 43).

[38] Aitao Guo, Aijun Liu, and Xiaodong Teng. “The pathology of urinary bladder lesions with an inverted growth pattern”. In: *Chinese Journal of Cancer Research* 28.1 (2016), p. 107 (cit. on p. 72).

[39]Clyde L Hardin, CL Hardin, and Luisa Maffi. *Color categories in thought and language*. Cambridge University Press, 1997 (cit. on p. 15).

[40]Mark Harrower and Cynthia A Brewer. "ColorBrewer.org: an online tool for selecting colour schemes for maps". In: *The Cartographic Journal* 40.1 (2003), pp. 27–37 (cit. on pp. 13, 15, 16, 125).

[41]Ernst Hellinger. "Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen." In: *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1909.136 (1909), pp. 210–271 (cit. on p. 86).

[42]Nathalie Henry, Jean-Daniel Fekete, and Michael J McGuffin. "Nodetrix: a hybrid visualization of social networks". In: *IEEE transactions on visualization and computer graphics* 13.6 (2007), pp. 1302–1309 (cit. on p. 122).

[43]Julia Herold, Christian Loyek, and Tim W Nattkemper. "Multivariate image mining". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1.1 (2011), pp. 2–13 (cit. on pp. 6, 39).

[44]Michiel de Hoon, Seiya Imoto, and Satoru Miyano. "The C clustering library". In: *Institute of Medical Science, Human Genome Center, University of Tokyo* 11 (2003) (cit. on p. 89).

[45]Harold Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of educational psychology* 24.6 (1933), p. 417 (cit. on p. 44).

[46]Mark D Humphries and Kevin Gurney. "Network 'small-world-ness': a quantitative method for determining canonical network equivalence". In: *PloS one* 3.4 (2008) (cit. on p. 103).

[47]Robert William Gainer Hunt and Michael R Pointer. *Measuring colour*. John Wiley & Sons, 2011 (cit. on p. 172).

[48]Chanchala Kaddi, R Mitchell Parry, and May D Wang. "Hypergeometric similarity measure for spatial analysis in tissue imaging mass spectrometry". In: *2011 IEEE International Conference on Bioinformatics and Biomedicine*. IEEE. 2011, pp. 604–607 (cit. on p. 86).

[49]Daniel A Keim. "Information visualization and visual data mining". In: *IEEE transactions on Visualization and Computer Graphics* 8.1 (2002), pp. 1–8 (cit. on p. 14).

[50]Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. "Visual analytics: Scope and challenges". In: *Visual data mining*. Springer, 2008, pp. 76–90 (cit. on p. 11).

[51]Ivo Klinkert, Kamila Chughtai, Shane R Ellis, and Ron MA Heeren. "Methods for full resolution data exploration and visualization for large 2D and 3D mass spectrometry imaging datasets". In: *International Journal of Mass Spectrometry* 362 (2014), pp. 40–47 (cit. on p. 39).

[52]Richard Knochenmuss. "Ion formation mechanisms in UV-MALDI". In: *Analyst* 131.9 (2006), pp. 966–986 (cit. on p. 4).

[53] Teuvo Kohonen. “Self-organized formation of topologically correct feature maps”. In: *Biological cybernetics* 43.1 (1982), pp. 59–69 (cit. on p. 144).

[54] Jan Kölling, Daniel Langenkämper, Sylvie Abouna, Michael Khan, and Tim W Nattkemper. “WHIDE—a web tool for visual data mining colocation patterns in multivariate bioimages”. In: *Bioinformatics* 28.8 (2012), pp. 1143–1150 (cit. on pp. xii, 9, 143, 167, 169).

[55] Jan Kölling, Martin Zurowietz, and Tim Wilhelm Nattkemper. “Interactive Exploration of Spatial Distribution in Mass Spectrometry Imaging”. Presented at the 22nd Annual International Conference on Intelligent Systems for Molecular Biology (ISMB). 2014 (cit. on p. 180).

[56] Peter Kovesi. “Good colour maps: How to design them”. In: *arXiv preprint arXiv:1509.03700* (2015) (cit. on p. 178).

[57] Andrea Lancichinetti and Santo Fortunato. “Community detection algorithms: a comparative analysis”. In: *Physical review E* 80.5 (2009), p. 056117 (cit. on p. 114).

[58] Andrea Lancichinetti and Santo Fortunato. “Consensus clustering in complex networks”. In: *Scientific reports* 2 (2012), p. 336 (cit. on p. 137).

[59] Vito Latora and Massimo Marchiori. “Efficient behavior of small-world networks”. In: *Physical review letters* 87.19 (2001), p. 198701 (cit. on p. 104).

[60] Haim Levkowitz. “Color icons-merging color and texture perception for integrated visualization of multiple parameters”. In: *1991 Proceeding Visualization*. IEEE Computer Society. 1991, pp. 164–170 (cit. on p. 169).

[61] Tony Lindeberg. “Scale-space”. In: *Wiley Encyclopedia of Computer Science and Engineering* (2007), pp. 2495–2504 (cit. on p. 74).

[62] Tony Lindeberg. “Scale-space theory: A basic tool for analyzing structures at different scales”. In: *Journal of applied statistics* 21.1-2 (1994), pp. 225–270 (cit. on p. 74).

[63] Markus Lux, Jan Krüger, Christian Rinke, et al. “acdc–Automated Contamination Detection and Confidence estimation for single-cell genome data”. In: *BMC bioinformatics* 17.1 (2016), p. 543 (cit. on p. 45).

[64] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605 (cit. on pp. 22, 44).

[65] Ernst Mach. “Über die Wirkung der räumlichen Vertheilung des Lichtreizes auf die Netzhaut”. In: *Sitzungsberichte der mathematisch-naturwissenschaftlichen Classe der kaiserlichen Akademic der Wissenschaftlen* 52 (1865), pp. 303–322 (cit. on p. 17).

[66] Alexander Makarov. “Electrostatic axially harmonic orbital trapping: a high-performance technique of mass analysis”. In: *Analytical chemistry* 72.6 (2000), pp. 1156–1162 (cit. on p. 6).

[67] BA Mamyrin. “Time-of-flight mass spectrometry (concepts, achievements, and prospects)”. In: *International Journal of Mass Spectrometry* 206.3 (2001), pp. 251–266 (cit. on p. 5).

[68]BA Mamyrin, VI Karataev, DV Shmikk, and VA Zagulin. “The mass-reflectron, a new nonmagnetic time-of-flight mass spectrometer with high resolution”. In: *Zh. Eksp. Teor. Fiz* 64.1 (1973), pp. 82–89 (cit. on p. 5).

[69]Pablo Marquez-Neila, Luis Baumela, and Luis Alvarez. “A morphological approach to curvature-based evolution of curves and surfaces”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.1 (2013), pp. 2–17 (cit. on p. 138).

[70]Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. “Robust wide-baseline stereo from maximally stable extremal regions”. In: *Image and vision computing* 22.10 (2004), pp. 761–767 (cit. on p. 138).

[71]Paul M Mather. “Cluster analysis for researchers”. In: *Computers and Geosciences* 18 (1992), pp. 98–98 (cit. on p. 69).

[72]*Matplotlib Colormaps*. `https://bids.github.io/colormap/`. [Online; accessed 12-September-2019] (cit. on pp. 17, 125).

[73]Gregor McCombie, Dieter Staab, Markus Stoeckli, and Richard Knochenmuss. “Spatial and spectral correlations in MALDI mass spectrometry images by clustering and multivariate analysis”. In: *Analytical chemistry* 77.19 (2005), pp. 6118–6124 (cit. on pp. 9, 143, 182).

[74]Liam A McDonnell and Ron MA Heeren. “Imaging mass spectrometry”. In: *Mass spectrometry reviews* 26.4 (2007), pp. 606–643 (cit. on p. 2).

[75]Leland McInnes, John Healy, and James Melville. “Umap: Uniform manifold approximation and projection for dimension reduction”. In: *arXiv preprint arXiv:1802.03426* (2018) (cit. on pp. 22, 37, 44).

[76]Wes McKinney et al. “Data structures for statistical computing in python”. In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56 (cit. on p. 43).

[77]Kenneth Moreland. “Diverging color maps for scientific visualization (expanded)”. In: *Proceedings in ISVC 9* (), pp. 1–20 (cit. on pp. 15, 16).

[78]Tamara Munzner. *Visualization analysis and design*. CRC press, 2014, pp. 135–142 (cit. on pp. 12, 14).

[79]Tamara Munzner. *Visualization analysis and design*. CRC press, 2014, pp. 299–321 (cit. on p. 13).

[80]Tamara Munzner. *Visualization analysis and design*. CRC press, 2014, pp. 219–241 (cit. on pp. 14–16).

[81]Mark EJ Newman. “Modularity and community structure in networks”. In: *Proceedings of the national academy of sciences* 103.23 (2006), pp. 8577–8582 (cit. on pp. 114, 115).

[82]Mark EJ Newman and Michelle Girvan. “Finding and evaluating community structure in networks”. In: *Physical review E* 69.2 (2004), p. 026113 (cit. on p. 115).

[83]Andrei Novikov. “PyClustering: data mining library”. In: *Journal of Open Source Software* 4.36 (2019), p. 1230 (cit. on p. 89).

[84]Jörg Ontrup and Helge Ritter. “Large-scale data exploration with the hierarchically growing hyperbolic SOM”. In: *Neural networks* 19.6-7 (2006), pp. 751–761 (cit. on pp. 145–147).

[85]Nobuyuki Otsu. “A threshold selection method from gray-level histograms”. In: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66 (cit. on p. 90).

[86]Katja Ovchinnikova, Lachlan Stuart, Alexander Rakhlin, Sergey Nikolenko, and Theodore Alexandrov. “ColocML: machine learning quantifies co-localization between mass spectrometry images”. In: *Bioinformatics* 36.10 (2020), pp. 3215–3224 (cit. on pp. 69, 96, 97).

[87]Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. “Uncovering the overlapping community structure of complex networks in nature and society”. In: *nature* 435.7043 (2005), pp. 814–818 (cit. on p. 114).

[88]Andrew Palmer, Dennis Trede, and Theodore Alexandrov. “Where imaging mass spectrometry stands: here are the numbers”. In: *Metabolomics* 12.6 (2016), p. 107 (cit. on pp. 3, 4).

[89]Karl Pearson. “LIII. On lines and planes of closest fit to systems of points in space”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572 (cit. on p. 44).

[90]Vincent Picaud, Jean-Francois Giovannelli, Caroline Truntzer, et al. “Linear MALDI-ToF simultaneous spectrum deconvolution and baseline removal”. In: *BMC bioinformatics* 19.1 (2018), p. 123 (cit. on p. 50).

[91]Ronald M Pickett and Georges G Grinstein. “Iconographic displays for visualizing multidimensional data”. In: *Proceedings of the 1988 IEEE Conference on Systems, Man, and Cybernetics*. Vol. 514. 1988, p. 519 (cit. on p. 169).

[92]Josien PW Pluim, JB Antoine Maintz, and Max A Viergever. “Mutual-information-based registration of medical images: a survey”. In: *IEEE transactions on medical imaging* 22.8 (2003), pp. 986–1004 (cit. on p. 87).

[93]Alan M Race and Josephine Bunch. “Optimisation of colour schemes to accurately display mass spectrometry imaging data based on human colour perception”. In: *Analytical and bioanalytical chemistry* 407.8 (2015), pp. 2047–2054 (cit. on pp. 17, 18, 63).

[94]Alan M Race, Andrew D Palmer, Alex Dexter, et al. “SpectralAnalysis: software for the masses”. In: *Analytical chemistry* 88.19 (2016), pp. 9451–9458 (cit. on p. 39).

[95]Pere Ràfols, Sònia Torres, Noelia Ramírez, et al. “rMSI: an R package for MS imaging data handling and visualization”. In: *Bioinformatics* 33.15 (2017), pp. 2427–2428 (cit. on p. 39).

[96]Hadi Rezaeilouyeh, Ali Mollahosseini, and Mohammad H Mahoor. “Microscopic medical image classification framework via deep learning and shearlet transform”. In: *Journal of Medical Imaging* 3.4 (2016), p. 044501 (cit. on p. 72).

[97]Penny L Rheingans. “Task-based color scale design”. In: *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*. Vol. 3905. International Society for Optics and Photonics. 2000, pp. 35–43 (cit. on pp. 15, 16, 125).

[98]Bernice E Rogowitz, Lloyd A Treinish, and Steve Bryson. “How not to lie with visualization”. In: *Computers in Physics* 10.3 (1996), pp. 268–273 (cit. on pp. 16, 17, 125).

[99]Andreas Römpp, Sabine Guenther, Yvonne Schober, et al. “Histology by mass spectrometry: label-free tissue characterization obtained from high-accuracy bioanalytical imaging”. In: *Angewandte chemie international edition* 49.22 (2010), pp. 3834–3838 (cit. on pp. 5, 30).

[100]Peter J Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65 (cit. on pp. 78, 110).

[101]Oliver Rubel, Annette Greiner, Shreyas Cholia, et al. “OpenMSI: a high-performance web-based platform for mass spectrometry imaging”. In: *Analytical chemistry* 85.21 (2013), pp. 10354–10361 (cit. on p. 63).

[102]Mikail Rubinov and Olaf Sporns. “Complex network measures of brain connectivity: uses and interpretations”. In: *Neuroimage* 52.3 (2010), pp. 1059–1069 (cit. on p. 130).

[103]Danielle L Schmitt and Songon An. “Spatial organization of metabolic enzyme complexes in cells”. In: *Biochemistry* 56.25 (2017), pp. 3184–3196 (cit. on p. 2).

[104]Thomas D Schneider and R Michael Stephens. “Sequence logos: a new way to display consensus sequences”. In: *Nucleic acids research* 18.20 (1990), pp. 6097–6100 (cit. on p. 169).

[105]Thorsten Schramm, Alfons Hester, Ivo Klinkert, et al. “imzML—a common data format for the flexible exchange and processing of mass spectrometry imaging data”. In: *Journal of proteomics* 75.16 (2012), pp. 5106–5110 (cit. on p. 39).

[106]Hamid R Sheikh, Alan C Bovik, and Gustavo De Veciana. “An information fidelity criterion for image quality assessment using natural scene statistics”. In: *IEEE Transactions on image processing* 14.12 (2005), pp. 2117–2128 (cit. on p. 69).

[107]Jianbo Shi and Jitendra Malik. “Normalized cuts and image segmentation”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000), pp. 888–905 (cit. on p. 89).

[108]Ben Shneiderman. “The eyes have it: A task by data type taxonomy for information visualizations”. In: *Proceedings 1996 IEEE symposium on visual languages*. IEEE. 1996, pp. 336–343 (cit. on p. 12).

[109]Eero P Simoncelli and Bruno A Olshausen. “Natural image statistics and neural representation”. In: *Annual review of neuroscience* 24.1 (2001), pp. 1193–1216 (cit. on p. 69).

[110]Martin Slawski, Rene Hussong, Andreas Tholey, et al. “Isotope pattern deconvolution for peptide mass spectrometry by non-negative least squares/least absolute deviation template matching”. In: *BMC bioinformatics* 13.1 (2012), p. 291 (cit. on p. 50).

[111]Tina Smets, Nico Verbeeck, Marc Claesen, et al. “Evaluation of distance metrics and spatial autocorrelation in uniform manifold approximation and projection applied to mass spectrometry imaging data”. In: *Analytical chemistry* 91.9 (2019), pp. 5706–5714 (cit. on p. 44).

[112]Robert R Sokal. “A statistical method for evaluating systematic relationships”. In: *Univ. Kansas, Sci. Bull.* 38 (1958), pp. 1409–1438 (cit. on p. 88).

[113]Anuj Srivastava, Ann B Lee, Eero P Simoncelli, and S-C Zhu. “On advances in statistical modeling of natural images”. In: *Journal of mathematical imaging and vision* 18.1 (2003), pp. 17–33 (cit. on p. 69).

[114]Michael Steinbach, Levent Ertöz, and Vipin Kumar. “The challenges of clustering high dimensional data”. In: *New directions in statistical physics*. Springer, 2004, pp. 273–309 (cit. on p. 35).

[115]Lee J Sweetlove and Alisdair R Fernie. “The spatial organization of metabolism within the plant cell”. In: *Annual Review of Plant Biology* 64 (2013), pp. 723–746 (cit. on p. 2).

[116]Aditya Tandon, Aiiad Albeshri, Vijey Thayananthan, Wadee Alhalabi, and Santo Fortunato. “Fast consensus clustering in complex networks”. In: *Physical Review E* 99.4 (2019), p. 042301 (cit. on p. 137).

[117]The pandas development team. *pandas-dev/pandas: Pandas*. Version 0.25.3. Feb. 2020 (cit. on p. 43).

[118]William H Tedford Jr, SL Bergquist, and William E Flynn. “The size-color illusion”. In: *The Journal of General Psychology* 97.1 (1977), pp. 145–149 (cit. on pp. 16, 17, 125).

[119]J-P Thiran and Benoit Macq. “Morphological feature extraction for the classification of digital images of cancerous tissues”. In: *IEEE Transactions on biomedical engineering* 43.10 (1996), pp. 1011–1020 (cit. on p. 72).

[120]James Thomas and Kristen A Cook. “Illuminating the path: The R&D agenda for visual analytics national visualization and analytics center”. In: *National Visualization and Analytics Center* (2005) (cit. on p. 11).

[121]Edward R Tufte. *The visual display of quantitative information*. Vol. 2. Graphics press Cheshire, CT, 2001, pp. 107–121 (cit. on p. 13).

[122]*UMAP Uniform Manifold Approximation and Projection for Dimension Reduction | SciPy 2018 |*. `https://www.youtube.com/watch?v=nq6iPZVUxZU`. [Online; accessed 07-April-2020] (cit. on p. 37).

[123]*Understanding UMAP*. `https://pair-code.github.io/understanding-umap/`. [Online; accessed 07-April-2020] (cit. on p. 37).

[124]Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. “The NumPy array: a structure for efficient numerical computation”. In: *Computing in Science & Engineering* 13.2 (2011), p. 22 (cit. on p. 47).

[125]Kirill Veselkov, Jonathan Sleeman, Emmanuelle Claude, et al. “BASIS: High-performance bioinformatics platform for processing of large-scale mass spectrometry imaging data in chemically augmented histology”. In: *Scientific reports* 8.1 (2018), p. 4053 (cit. on pp. 39, 41, 42).

[126]Juan Antonio Vizcaíno, Richard G Côté, Attila Csordas, et al. “The PRoteomics IDEntifications (PRIDE) database and associated tools: status in 2013”. In: *Nucleic acids research* 41.D1 (2012), pp. D1063–D1069 (cit. on p. 31).

[127]Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, et al. “scikit-image: image processing in Python”. In: *PeerJ* 2 (June 2014), e453 (cit. on p. 139).

[128]Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612 (cit. on p. 69).

[129]Duncan J Watts and Steven H Strogatz. “Collective dynamics of ‘small-world’networks”. In: *nature* 393.6684 (1998), p. 440 (cit. on pp. 104, 105, 107, 131).

[130]Marco Wehofsky, Ralf Hoffmann, Martin Hubert, and Bernhard Spengler. “Isotopic deconvolution of matrix-assisted laser desorption/ionization mass spectra for substance-class specific analysis of complex samples”. In: *European Journal of Mass Spectrometry* 7.1 (2001), pp. 39–46 (cit. on p. 50).

[131]Patrick M Wehrli, Wojciech Michno, Kaj Blennow, Henrik Zetterberg, and Jörg Hanrieder. “Chemometric Strategies for Sensitive Annotation and Validation of Anatomical Regions of Interest in Complex Imaging Mass Spectrometry Data”. In: *Journal of The American Society for Mass Spectrometry* 30.11 (2019), pp. 2278–2288 (cit. on p. 139).

[132]Chalini D Wijetunge, Isaam Saeed, Saman K Halgamuge, Berin Boughton, and Ute Roessner. “Unsupervised learning for exploring MALDI imaging mass spectrometry ‘omics’ data”. In: *7th International Conference on Information and Automation for Sustainability*. IEEE. 2014, pp. 1–6 (cit. on p. 8).

[133]Mathias Wilhelm, Marc Kirchner, Judith AJ Steen, and Hanno Steen. “mz5: space- and time-efficient storage of mass spectrometry data sets”. In: *Molecular & Cellular Proteomics* 11.1 (2012) (cit. on p. 43).

[134]Andrew P Witkin. “Scale-space filtering”. In: *Readings in Computer Vision*. Elsevier, 1987, pp. 329–332 (cit. on p. 74).

[135]Bang Wong. *Negative space*. 2010 (cit. on p. 13).

[136]Bang Wong. *Points of view: Color coding*. 2010 (cit. on pp. 16, 125).

[137]Bang Wong. *Points of view: Design of data figures*. 2010 (cit. on p. 16).

[138]Bang Wong. *Points of view: salience to relevance*. 2011 (cit. on p. 14).

[139]Bang Wong. *Points of view: simplify to clarify*. 2011 (cit. on p. 13).

[140]Karsten Wüllems. “Exploration of Spatial Patterns in Bioimages using Community Detection”. Poster at OurCon V. 2017 (cit. on p. xii).

[141] Karsten Wüllems, Jan Kölling, Hanna Bednarz, et al. “Detection and visualization of communities in mass spectrometry imaging data”. In: *BMC bioinformatics* 20.1 (2019), p. 303 (cit. on p. xii).

[142] Karsten Wüllems and Tim W. Nattkemper. “SoRC – Evaluation of Computational Molecular Co-Localization Analysis in Mass Spectrometry Images”. In: (2020). arXiv: 2009.14677 [cs.CV] (cit. on pp. xi, xii).

[143] Karsten Wüllems, Annika Zurowietz, Martin Zurowietz, et al. “Fast and Visual Exploration of Mass Spectrometry Images with Interactive Dynamic Spectral Similarity Pseudocoloring”. In: *Scientific reports* (in revision) (cit. on pp. xi, xiii, 56, 57).

[144] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. “Overlapping community detection in networks: The state-of-the-art and comparative study”. In: *Acm computing surveys (csur)* 45.4 (2013), pp. 1–35 (cit. on p. 114).

[145] Fuyong Xing and Lin Yang. “Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review”. In: *IEEE reviews in biomedical engineering* 9 (2016), pp. 234–263 (cit. on p. 72).

[146] Chao Yang, Zengyou He, and Weichuan Yu. “Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis”. In: *BMC bioinformatics* 10.1 (2009), p. 4 (cit. on p. 50).

[147] Zhao Yang, René Algesheimer, and Claudio J Tessone. “A comparative analysis of community detection algorithms on artificial networks”. In: *Scientific reports* 6 (2016), p. 30750 (cit. on p. 114).

[148] Ji Soo Yi, Youn ah Kang, and John Stasko. “Toward a deeper understanding of the role of interaction in information visualization”. In: *IEEE transactions on visualization and computer graphics* 13.6 (2007), pp. 1224–1231 (cit. on p. 14).

[149] Yune Yuan, Qing Wang, Ji Ye Li, Zhongqi Liu, et al. “Image analysis of breast tumors using thermal texture mapping (TTM)”. In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE. 2006, pp. 697–699 (cit. on p. 72).

[150] Gergely Zahoránszky-Kőhalmi, Cristian G Bologa, and Tudor I Oprea. “Impact of similarity threshold on the topology of molecular similarity networks and clustering outcomes”. In: *Journal of cheminformatics* 8.1 (2016), p. 16 (cit. on pp. 103, 105, 106).

List of Figures

List of Tables

Symbols

$\mathcal{I}$ MSI data set

$\mathcal{I}_{h,w}$ A mass spectrum of a specified position

$\mathcal{I}_z$ An m/z-image of a specified m/z-values

$\mathcal{I}_{h,w,z}$ A specified single intensity value

p Total number of all pixel position tuples

ρ Total number of all multivariate spectral pixel position tuples

H Total number of pixel rows

h Running index of pixel rows

W Total number of pixel columns

w Running index of pixel columns

Z Total number of m/z-values

z Running index of m/z-values

k Number of clusters

$|\cdot|$ Size of a set

d Dimensionality of a data set

CC Connected component

ϕ Peak picking threshold

ϕ^+ Upper deisotoping limit

ϕ^- Lower deisotoping limit

μ Arithmetic mean function

σ Standard deviation function

$\mathcal{I}^\nabla$ Vector map

$\mathcal{I}^{\mathrm{M}}$ Magnitude image

$\mathcal{I}^{\mathrm{G}}$ Gradient image

$\mathcal{I}^{\mathrm{V}}$ Vectorized image

$\mathcal{I}^{\mathrm{P}}$ Image histogram

Υ Set of all distances resulting from cluster merges of hierarchical clustering

ς Cophenetic distance threshold

ψ SoRC score

$\mathcal{F}$ Set of similarity functions

$\mathcal{P}$ Set of pre-processed data sets

S Set of scale-space levels

s Scale-space level step size

t Scale-space level

$\mathfrak{C}$ Result of a clustering method

g Two-dimensional Gaussian kernel

L Convolution function

κ Sliding window function

Ξ A similarity map

$\mathfrak{B}$ Sum of over a binarized image

HD Hellinger distance function

ι Histogram binning

H Entropy

$\mathfrak{U}$ Contingency table function

$\mathcal{U}$ Contingency table

θ Standard deviation kernel

V Set of vertices

$\mathfrak{V}_i$ Local neighborhood of vertex v_i

v_i Vertex

E Set of edges

e_i Edge

G Graph

g_i Sub-graph

$\mathfrak{p}$ Path in a graph

$\mathcal{S}$ Similarity matrix

$\mathcal{A}$ Adjacency matrix

Γ Set of candidate thresholds

γ_i Candidate threshold

Λ_c Average clustering coefficient

$\hat{\Lambda}_c$ Normalized average clustering coefficient

Λ_e Average efficiency

$\hat{\Lambda}_e$ Normalized average efficiency

$\mathcal{Q}$ Matrix consisting of the average clustering coefficient and the average efficiency

$\mathcal{Y}$ Matrix of sorted eigenvectors

Ψ Self-organizing map

u Neural unit/prototype

b Neural unit/prototype position

τ Neigborhood bell function for SOM training

$\mathfrak{w}$ Width of the neigborhood bell function for SOM training

$\mathfrak{b}$ Beam search beam-width

ε SOM training learning rate

A Maximum number of SOM training iterations

a SOM training iteration

$\mathfrak{r}$ H^2SOM ring index

$\mathfrak{R}$ Maximum number of H^{2}SOM rings

$\mathfrak{n}$ Number of neighbors in the H^{2}SOM ring

ϑ Stop criterion of position optimization

$\mathfrak{f}$ Position optimization mitigation factor

$\mathfrak{c}$ Position optimization change rate

$\bar{b}$ Approximated prototype position

$\mathcal{M}$ Colormap function

η Color value

η' Inverse angular distance

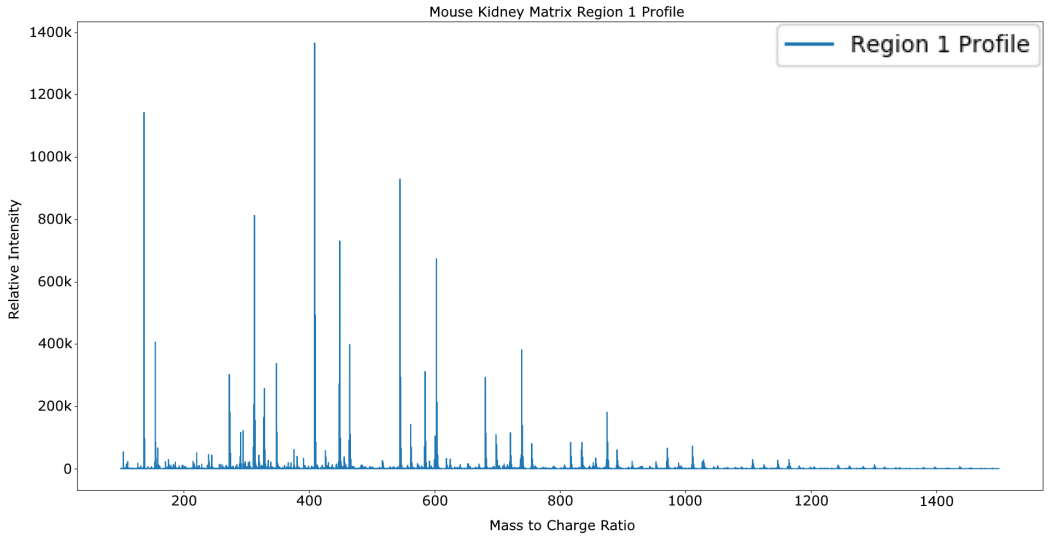
η'' Normalized inverse angular distance

$\widetilde{\mathcal{I}}$ 8-bit normalized MSI data cube

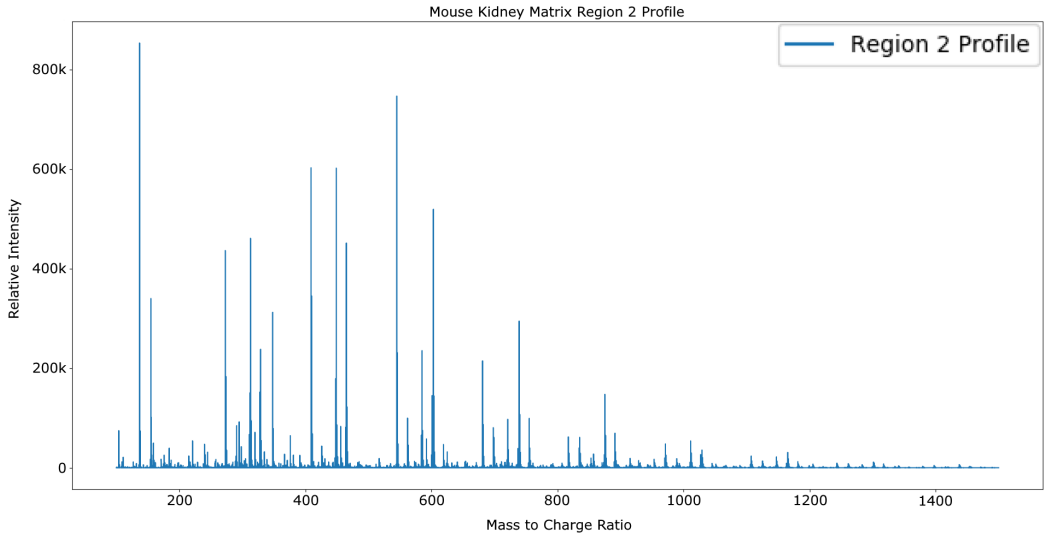
δ Generic distance function

Appendix A

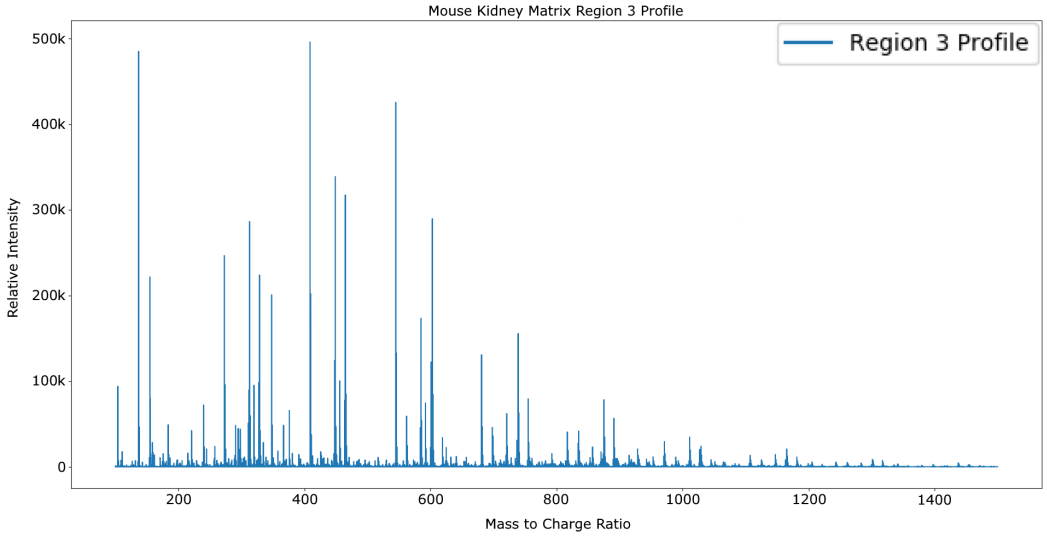
A.1 ProViM Mass Spectra Comparison



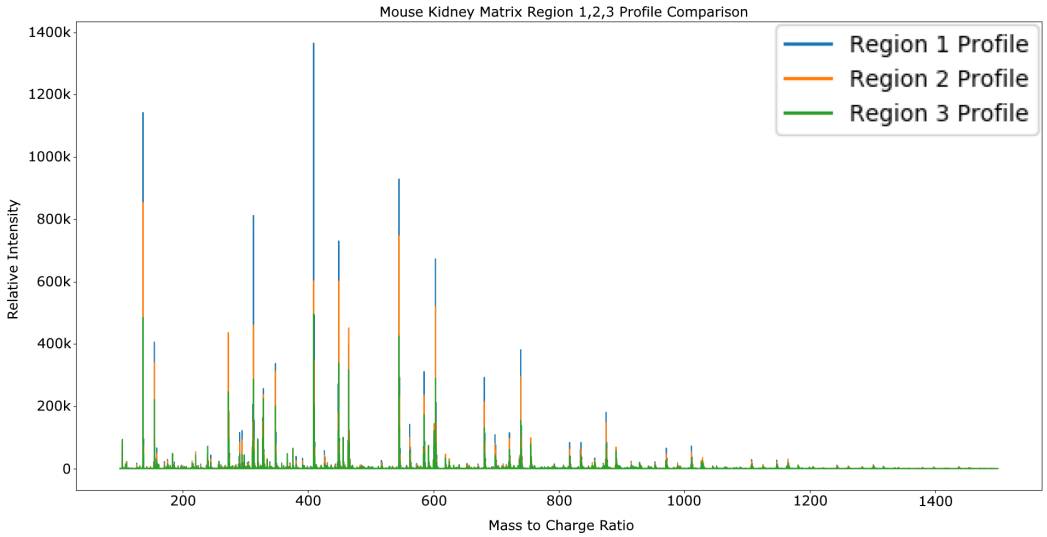
(A) Profile of the small group of matrix pixels



(B) Profile of the medium group of matrix pixels

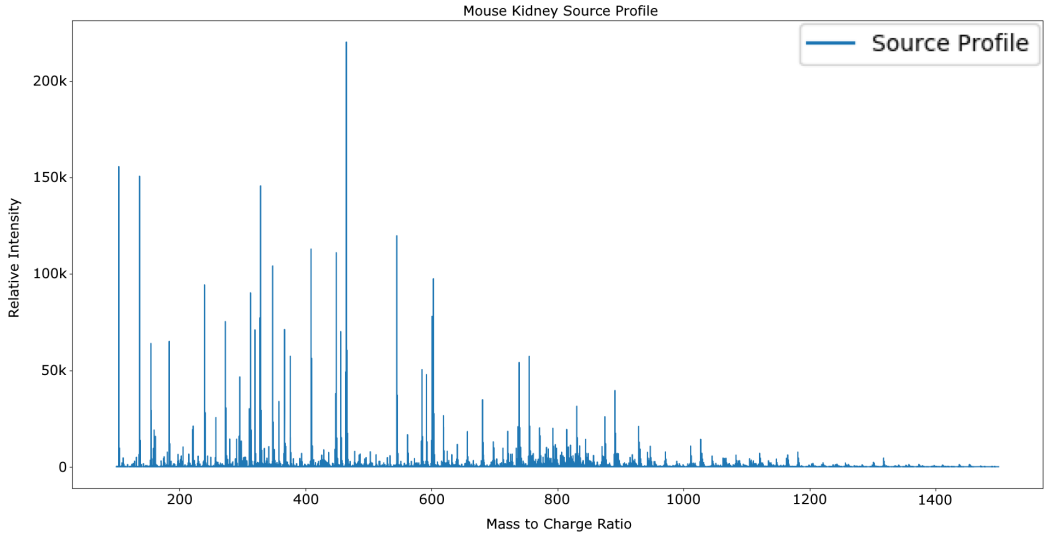


(C) Profile of the large (branching) group of matrix pixels

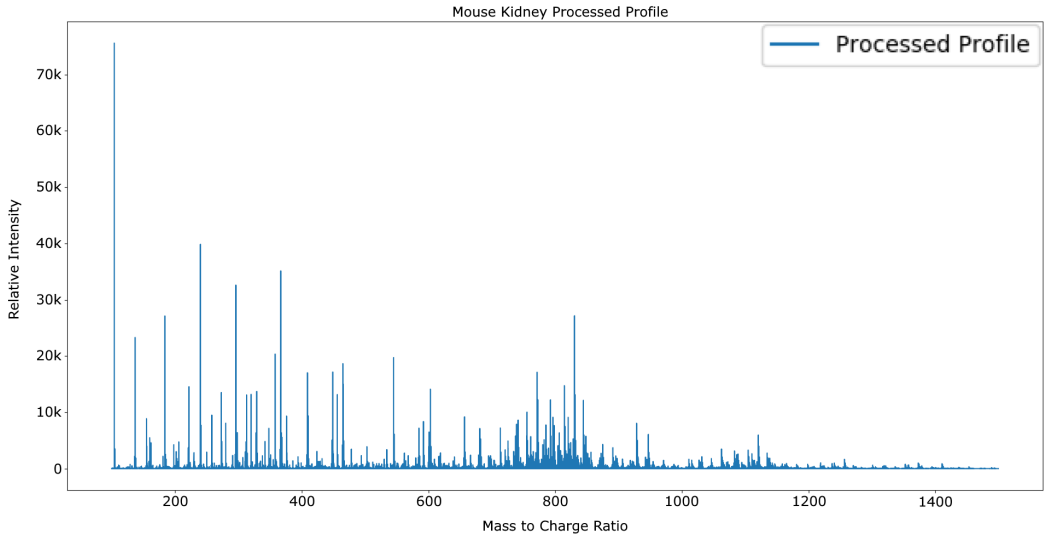


(D) Superimposition of (A),(B) and (C)

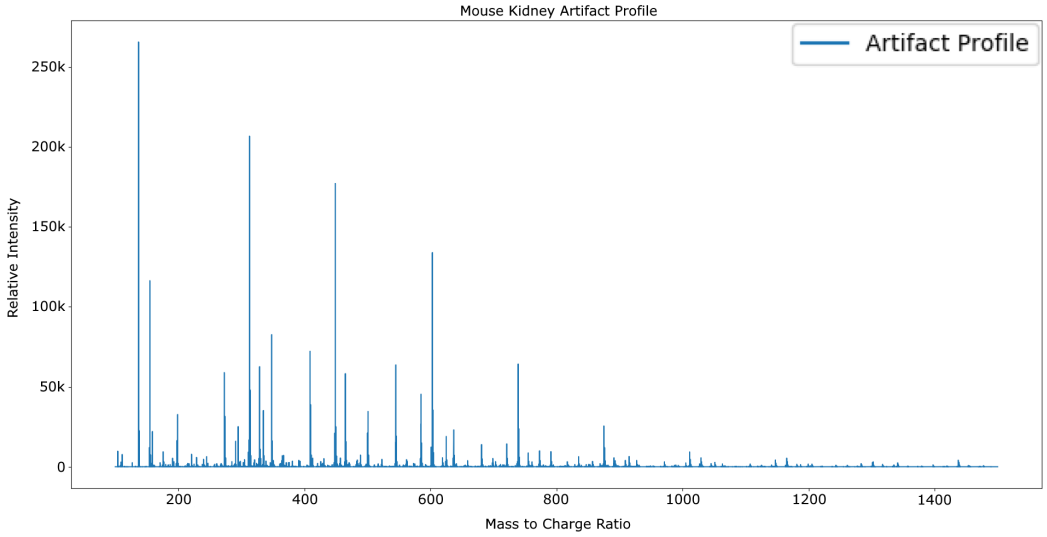
Fig. A.1.: Matrix profile comparison of the different subsets – (A), (B) and (C) show the matrix profiles of the small, medium and large pixel subset of Figure 3.6. It can be seen that the profiles show some differences in their spectral pattern and overall magnitude. However, the direct comparison in (D) shows that all regions share a similar basic profile. Profile calculation was done after the removal of all mass spectra located at $\mathcal{I}^{\mathrm{K}}_{0,0}$.



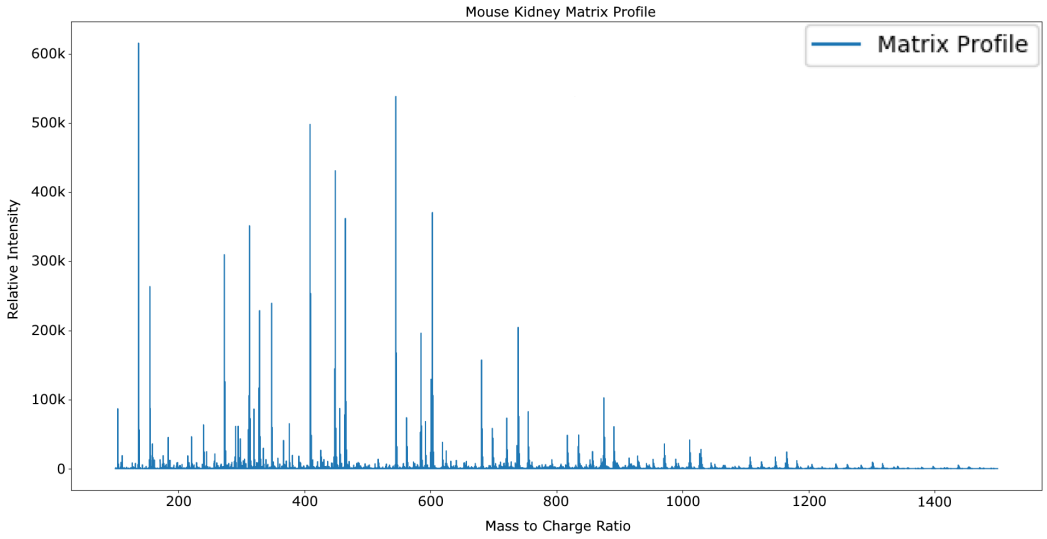
(A) Data set profile spectrum before MArDeR



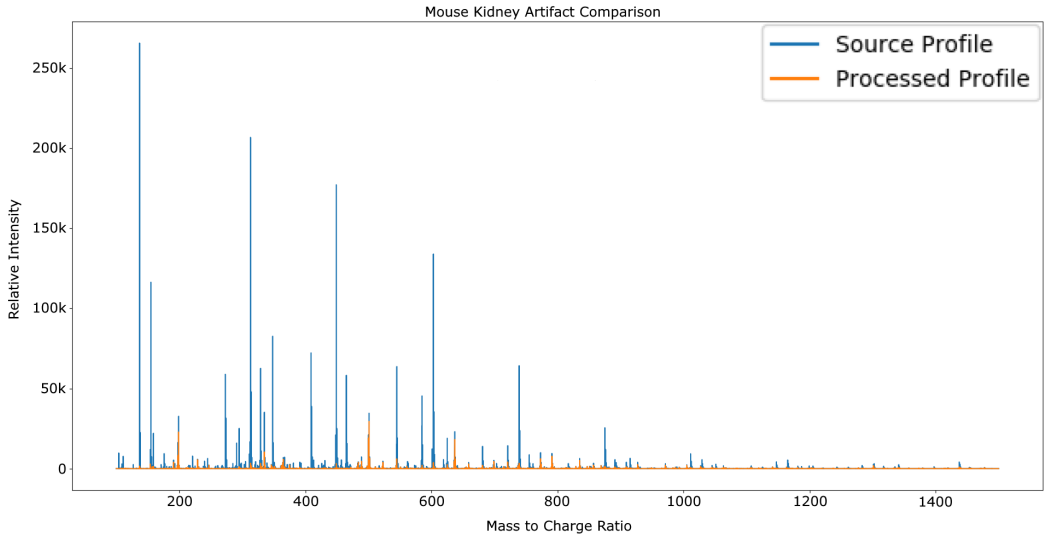
(B) Data set profile spectrum after MArDeR



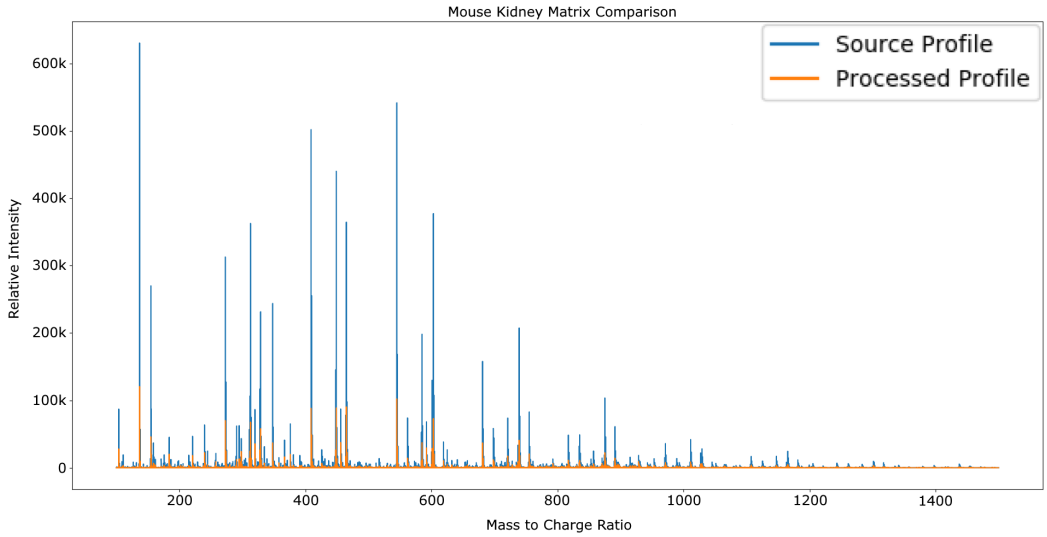
(C) Artifact profile spectrum



(D) Matrix profile spectrum



(E) Artifact profile comparison before (**blue**) and after (**orange**) MArDeR processing



(F) Matrix profile comparison before (**blue**) and after (**orange**) MArDeR processing

Fig. A.2.: Comparison of spectral profiles – All profiles were computed after the removal of all mass spectra located at $\mathcal{I}^{\mathrm{K}}_{0,0}$.

Colophon

This thesis was typeset with LaTeX 2_ε. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.

Declaration

I hereby declare that all the work and ideas presented in this thesis are my own or that I have otherwise marked and/or cited all other ideas and the sources I rely on.

Schwalmtal, November 5, 2020

Karsten Wüllems