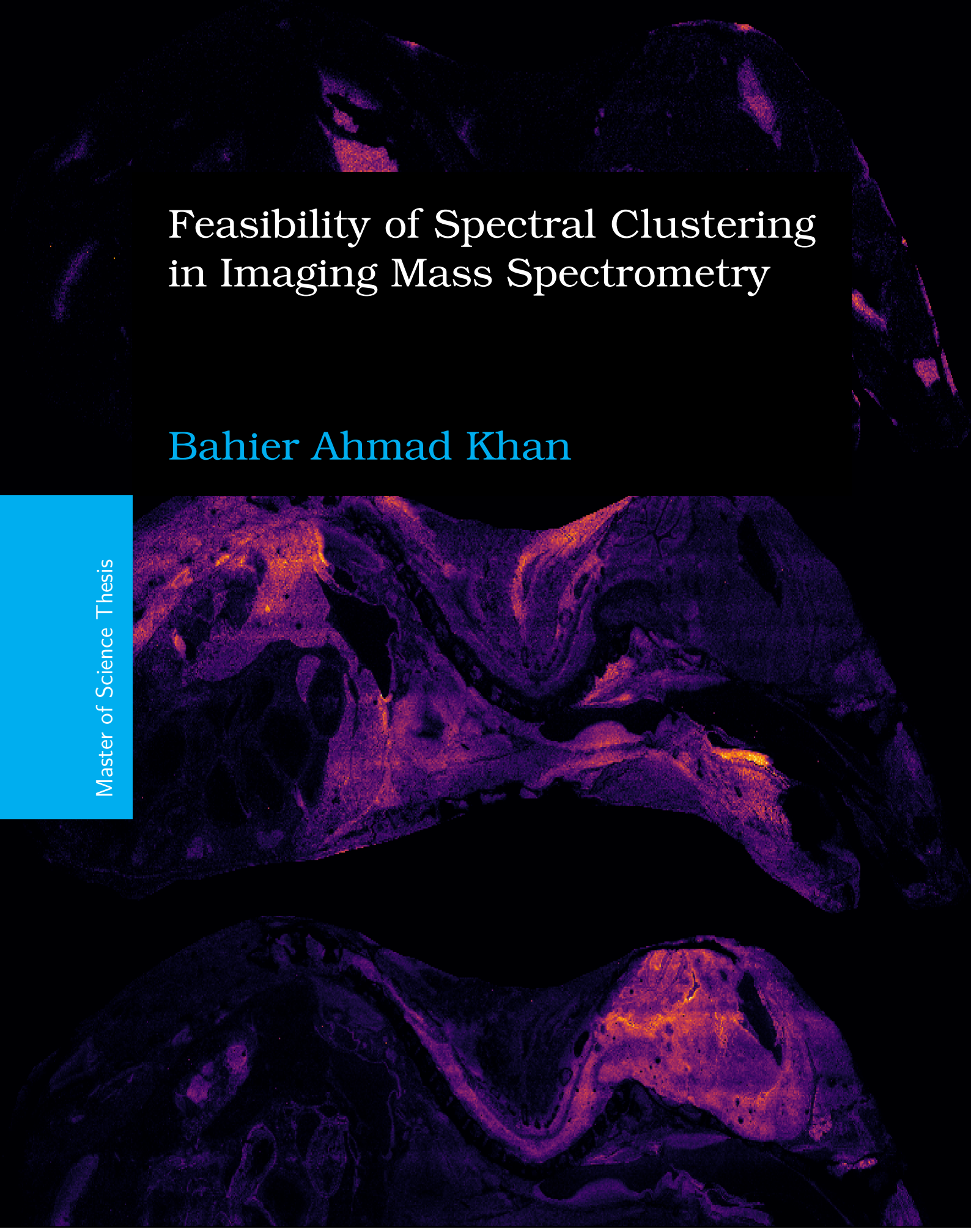
Feasibility of Spectral Clustering in Imaging Mass Spectrometry

Bahier Ahmad Khan



Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Feasibility of Spectral Clustering in Imaging Mass Spectrometry

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Bahier Ahmad Khan

October 19, 2025

Faculty of Mechanical Engineering (ME) · Delft University of Technology

Abstract

Imaging Mass Spectrometry (IMS) collects spatial and chemical information of a sample, generating high-dimensional datasets that present challenges in exploratory data analysis due to their substantial size. Spectral clustering is a promising unsupervised learning approach for IMS applications, employing graph-based strategies to identify patterns without assumptions about cluster geometry.

Unlike many clustering algorithms that have assumptions about the geometry of the clusters, spectral clustering constructs a similarity graph and performs eigendecomposition on the Laplacian matrix to reveal non-convex clusters. This allows one to find clusters of arbitrary shapes, which can result in new or improved segmentation being discovered in IMS data. Furthermore, two recent studies allow for the potential argument that spectral clustering might be optimal for IMS data.

Despite these advantages, spectral clustering faces implementation barriers primarily due to its computational complexity and memory constraints. The limited applications of spectral clustering on IMS data, can predominantly be attributed to these limitations.

This thesis investigates the feasibility of spectral clustering for analyzing high-dimensional imaging mass spectrometry data, with a focus on performance under noise, computational scalability, and maintaining biological segmentation. To assess the performance, internal and external validation metrics are used as well as a comparison with variations of k-means clustering. Additionally, a memory constrained algorithm was developed to address the scalability issue induced by the memory complexity.

The results highlight that spectral clustering outperforms k-means, when both methods are utilizing the cosine metric, in scenarios of increased noise on a synthetic dataset. Upon application on a real world subset of an IMS dataset of a mouse pup containing the brain, the results between k-means and spectral clustering were highly comparable. When applied on the complete dataset with a memory constrained version of spectral clustering, the results were less promising due to its dependence on initial seeding, where k-means obtained better or similar clustering results with lower time and memory complexity.

Table of Contents

Acknowledgements

This thesis encapsulates my final work before earning my Master of Science, and it was quite the challenge. I would like to extend a huge thank you to my friends from my bachelor in aerospace engineering and my master in system and control for their support and their company during my time studying and the all the activities outside of it. I am especially grateful to my family, who always believed in me and provided me with support throughout my life. Finally, I would like to extend my gratitude to my daily supervisor, Ir. Paul-Louis Delacour, and to my supervisor, Dr. Ir. Raf Van de Plas, for their guidance and insight.

Delft, University of Technology
October 19, 2025

Bahier Ahmad Khan

Chapter 1

Introduction and Background

Imaging Mass Spectrometry (IMS) is an analytical technique that allows the retrieval of both chemical and spatial information of a sample. This technique maps the spatial organization of molecules directly from tissue section, single cells and various other surfaces without the need for prior extraction or labeling of target molecules [1, 2]. The molecular information encoded in IMS datasets makes it possible to automatically segment tissues into medically significant subregions, which is in particular interest for applications such as clinical or pathological applications [3, 4, 5] .

Due to the large size and complexity of IMS data, manual interpretation becomes impractical. To address this, researchers apply exploratory data analysis to understand the structure, patterns, relationships and anomalies within the data. This can be achieved through statistical or machine learning analyses [6].

Among these techniques, clustering belongs to the branch of unsupervised machine learning, where data points are grouped together based on how similar they are to each other without the presence of any prior labels. Clustering does not only aid in pattern recognition but also enables the discovery of molecular signatures and spatial domains.

In the context of IMS, clustering pixel locations by looking at the similarity in their mass spectra, groups regions with comparable molecular profiles. As a result, clustering enables the identification of distinct molecular regions within complex tissue structures. This segmentation generates a spatial molecular map that highlights regions of interest within the tissue sections revealing subregions with comparable compositions.

This chapter provides background information on IMS and clustering, covered in section 1-1 and section 1-2, respectively. This context lays the foundation for formulating the problem statement (section 1-3) and facilitates a better understanding of the rest of the report.

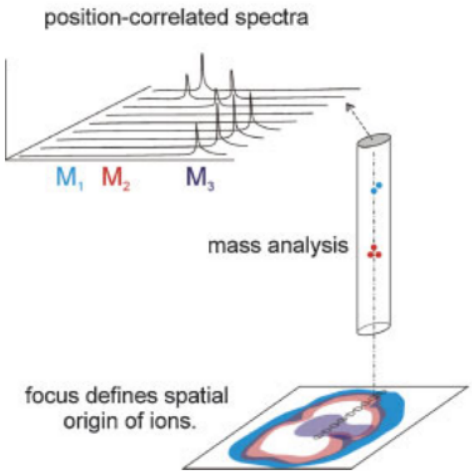
1-1 Introduction to Imaging Mass Spectrometry

The fundamental principle behind IMS involves the acquisition of the mass spectra from a sample surface along multiple predefined measurement locations. The obtained data can be

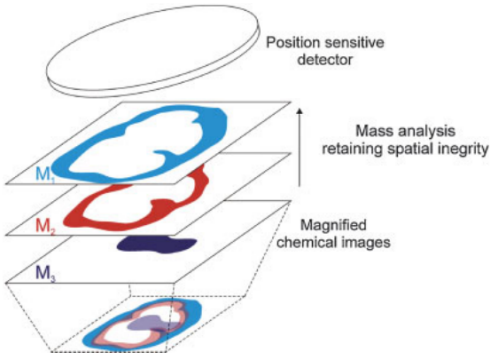
represented in a three-dimensional dataset where each pixel, corresponding to a measurement location, contains a complete mass spectrum. There are two approaches in which the data can be retrieved from a sample, these are denoted as microprobe mode imaging and microscope mode imaging [1, 2, 7].

Microprobe mode imaging This imaging mode uses a two-dimensional grid with measurement locations. At each of these measurement locations, the sample is subjected to an ionization beam, which vaporizes the tissue sample, changing its state from a solid/liquid phase to a gaseous one. The gas is fed into a mass analyzer where the analytes are characterized by their mass to charge ratio (m/z). This process is repeated over the complete grid to obtain a dataset for the complete sample, see Figure 1-1a.

Microscope mode imaging Instead of using pre-defined grid points, in microscope mode imaging a large area of the sample is subjected to an ionization beam. Whilst the sample has transitioned into the gas phase for a larger area, the use of specialized ion optics allow for the retention of their spatial distribution as they are extracted and travel through the mass analyzer. Specialized ion optics magnify and project this distribution onto a position-sensitive detector, see Figure 1-1b.



(a) The microprobe mode imaging can be seen going over prespecified locations and the gas is than fed into a mass analyzer.



(b) Different images of mass spectra are obtained by using the microscope mode imaging.

Figure 1-1: The two approaches in molecular imaging mass spectrometry. [Image from [2]]

The amount of pixels are determined by the achievable spatial resolution, which can reach into the order of 1 to 10 μm. The selected ionization technique determines how high the spatial resolution can be. There are many ionization techniques, such as Secondary Ion Mass Spectrometry (SIMS) [8], Desorption Electrospray Ionization (DESI) [9] or Matrix-Assisted Laser Desorption/Ionization (MALDI) [10]. We will focus on MALDI, as this is the technique used to obtain the data that will be analyzed.

In MALDI-IMS, the sample is coated in a matrix that absorbs the energy of the ionization beam. The matrix assists in minimizing the fragmentation when the ionization beam

is traversing the raster pattern. The technique allows for the preservation of the larger molecules, which results in high accuracy when used for imaging of proteins, peptides, lipid and metabolites.

The higher the spatial resolution, the more pixels are present in the obtained dataset, which can accumulate to datasets consisting of more than 100,000 pixels. Each of these data points has a discretised mass spectrum connected to it. The mass spectrum is organized into bins and each one often has hundreds of bins (Figure 3-1). This highlights the complex nature of IMS datasets, which will prove to be a challenge when trying to employ exploratory data analysis (see section 1-2).

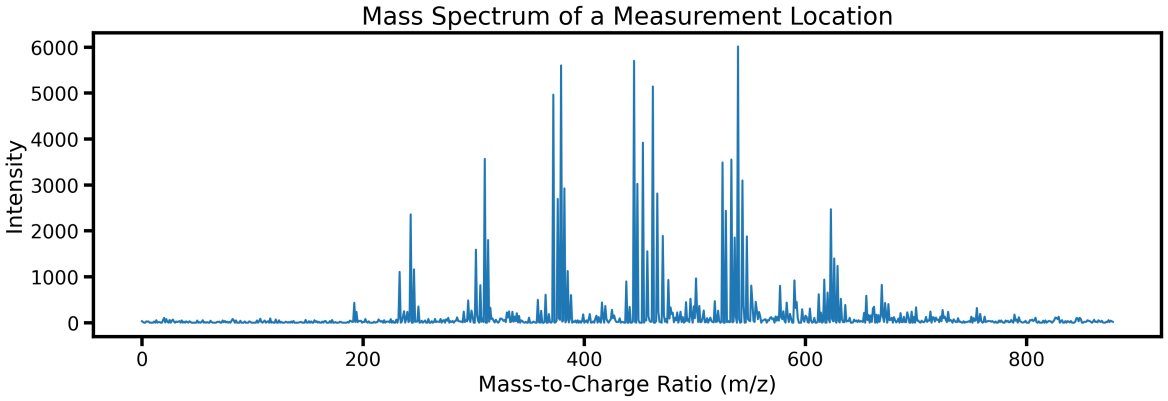


Figure 1-2: Example of a mass spectrum.

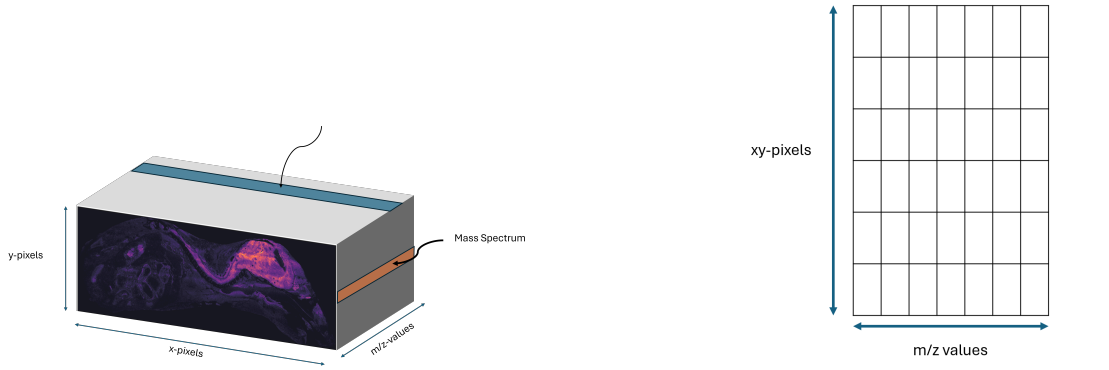
The collected data in imaging mass spectrometry can be represented in two primary formats; the 3D-mode data array and the 2D-mode data array (Figure 1-3). In the 3D-mode data array, the data is structured with three dimensions, the $x - y$ pixel directions which indicate the spatial locations where the data was collected on the sample in 2D space, and the third dimension which corresponds to the m/z bins [11]. This format allows for a comprehensive visualization of molecular distributions at the measurement locations of the sample.

Alternatively, a 2D-mode data array can be employed. Here, one axis represents the combined x and y pixel locations, effectively flattening the spatial information into a single dimension. The other axis remains dedicated to the m/z bins, capturing the spectral information. This representation simplifies the data structure but still retains the essential spatial and spectral relationships [2, 11].

Whilst the 3D mode data array is useful for visualizing individual ion images or mass spectra of a pixel, the 2D mode finds its use more in the analysis of the data.

From the aforementioned information, one can imagine that the data generated through the process of IMS can be large in size and complex. After all, a discrete signal is stored for each measurement location and for a good resolution of the tissue sample an adequate amount of measurement locations have to be chosen. The implications of this are notable when one would like to apply data analysis, which will be discussed in section 1-2.

Another problem with IMS data is that it contains noise [12, 13, 14]. Examples of noise are background noise, noise from the measurement equipment or noise due to incorrect sample



(a) Illustration of a 3D IMS dataset. **(b)** Illustration of a 2D IMS dataset.

Figure 1-3: The different data formats for IMS.

preparations leading to the presence of bubbles and dust. This noise prevents us from observing the true spectra, and should be filtered out. Noise can present itself as peaks, which can than be mistaken as the presence of a bio-chemical [12].

Even though it not being compulsory, preprocessing is often applied step when working with IMS data. Preprocessing consists of several steps to prepare the raw data for data analysis. Typical procedures may include smoothing, baseline correction and denoising, among others [15, 12]. These steps address common issues in raw spectra such as signal noise, baseline drift, and experimental variability.

The aim of pre-processing is to enhance desired features whilst minimizing artifacts, to ensure that the data reflects the tissue sample accurately. This phase allows for more effective analyses when clustering, and could improve interpretability and reproducibility of the results. The preprocessing step relevant for this thesis is total ion count normalization, as the provided dataset has already been processed from the raw file.

Total Ion Count (TIC) Normalization Given a mass spectrum ($\vec{s} = y_1, y_2, \ldots, y_n$), normalization is nothing more than,

$$s_{\vec{normalized}} = \frac{1}{f}\vec{s}, \text{ where } f = \left(\sum_i |y_i|^p\right)^{1/p}.$$

In the case $p = 1$, the mass spectrum is divided by the sum of all intensities and this is called TIC normalization [16]. This method has been applied commonly when looking at MALDI-IMS datasets. The idea is that this normalization will correct for variations in total signal intensity between different spectra. However, TIC is based on the assumption that the total ion count remains constant on average across samples and that signals do not decrease or increase in intensity systematically. This why the application of TIC could cause artifacts to occur and is why Deininger et al. recommends to use it after considering these possible implications [16].

1-2 Introduction to Clustering in Imaging Mass Spectrometry

As mentioned earlier, clustering is an unsupervised machine learning method that groups data points based on their similarity or distance with unlabeled data. In the field of IMS, clustering is applied in two different ways on the data.

Firstly, the data can be clustered based on the similarity in their mass spectra at each measurement location. This way the pixels of the image are clustered together providing a tissue segmentation map, which allows insight into anatomical regions. Therefore, this method is also called pixel-based clustering or spatial clustering/segmentation [6, 17, 18].

The second method regards ion image clustering, where the m/z values are clustered based on their spatial distribution across the tissue. This time relevant molecular ions are clustered that show similar spatial distribution patterns, which allows for the interpretation molecular pathways within the tissue section [19, 20].

Both methods of clustering IMS data have seen many applications. Each of these with different improvements and clustering algorithms accompanied. Some clustering algorithms include k-means [18, 21], hierarchical clustering [22, 23, 24], but also more learning based methods such as deep clustering [25, 26] .

This highlights the interest and need for improved clustering performance, as one clustering method can not be the best in every area. There are many challenges that show up when using clustering within IMS, some induced due to the complexity and size of the IMS data, whilst others are a consequence of the clustering algorithm and its unsupervised nature. These challenges will now be discussed, to provide the groundwork for when the problem statement is presented.

Curse of Dimensionality As stressed in section 1-1, IMS data is vast and complex. Firstly, depending on the application, the amount of pixels present in IMS data can range into an order of $10^4 - 10^6$, whilst the mass spectra at each pixel location can be discretized into bins in the range of $10^2 - 10^5$. The latter is of particular interest for clustering purposes.

Most clustering algorithms are developed with a specific metric in mind, be it either a distance or similarity. To reiterate, the data points are grouped based on the chosen distance or similarity metric, where data points with a smaller distance or higher similarity are often grouped together.

However, when the data has high dimensionality the distance between points tends to become relatively uniform [27, 28, 29], making the notion of close and far data points ambiguous. Therefore, the clustering algorithms will also deteriorate in performs.

In the field of IMS, feature selection and the projection of data onto lower dimensional spaces are used strategies to mitigate the effects of high dimensionality. Feature selection, essentially cleans up the data in a way that irrelevant features are discarded. There are many different ways in which this can be achieved and feature selection is a difficult topic by itself and considered outside the scope of this thesis.

For dimensionality reduction, the idea is that the complete dataset can be approximated by using a lower amount of dimensions. However, often the dimensions are still quite large even if they have been reduced significantly. This is because the amount of data needed to

cover the amount dimensions scales with an estimate of $O(c^d)$, where c is a constant and d the dimension of the data [30]. Therefore, even if it is reduced to a dimension of order $10^1 - 10^2$, the amount of data is still insufficient to give a proper coverage of the dimensional space. Nevertheless, many results have shown to be promising when applying dimensionality reduction techniques [6].

Computational Complexities When computing clustering results, IMS necessitates the a sufficient amount of memory for storing intermediate results and the high dimensionality of the data causes significantly more effort for the calculations of the distance or similarity metrics.

The memory requirements vary significantly between clustering algorithms. For instance, k-means has a memory complexity of $\mathcal{O}(nd)$, where n is the number of data points and d is the number of dimensions. In contrast, spectral clustering requires $\mathcal{O}(n^2)$ memory, as it involves computing and storing a full similarity matrix. Since in most practical scenarios $n \gg d$, spectral clustering typically demands substantially more memory than k-means.

Similarly, the time complexity of the algorithms is also impacted by the properties of the data. When looking again at k-means clustering the computational complexity is $\mathcal{O}(nkdi)$, where n is the number of points, k the number of clusters, d is the dimensionality and i is the number of iterations. Whilst, spectral clustering has a much higher computational complexity of $\mathcal{O}(n^3)$.

This shows that depending on the clustering algorithm, results can either be computed in relatively short or large amount of time. Therefore, when selecting an algorithm that has a high computational and memory cost, the benefits should be sufficient to substantiate the use of the algorithm.

Algorithmic Challenges Clustering algorithms inherently depend on a number of hyperparameters, with the most common one being the number of clusters k. These hyperparameters must be carefully selected, as they play an important role in determining the quality of the resulting clusters.

In particular, the way an algorithm is initialized, for example the selection of initial cluster centers in k-means, can have a significant impact on the final outcome.This sensitivity to initialization results in several challenges. It introduces inconsistency in the results, especially for algorithms that tend to to converging to local minima. Meaning, that the same algorithm could produce different clustering results when run multiple times with a different initialization of the centroids. To address this issue, it is common practice to run clustering algorithms multiple times with different initializations and to select the solution that optimizes a given objective function. Improved initialization techniques, such as k-means++, have been developed to increase the likelihood of convergence to a better solution by carefully selecting initial cluster centers [31].

Furthermore, determining an appropriate value for k is often non-trivial and is an issue shared by many clustering algorithms if not all. Some algorithms may benefit of the use of internal evaluation metrics like the silhouette score or the Davies–Bouldin index to assist in assessing clustering quality and guiding the choice for the number of clusters [18]. However, as these validation metrics are based on assumptions on the retrieved clustering results, it is not

an universal solution and in most cases running the algorithm with different values for the number of clusters is the best method.

Noise This challenge has relations to feature selection, given that the complete dataset does not contain only relevant information, but also several sources of noise. Separating the noise from the biologically relevant signals will help in obtaining better clustering results, as an increase in similarity through the inclusion of noise can be induced.

Validation As clustering is an unsupervised machine learning technique, it means that the data is unlabeled. This provides a particular challenge when validating results. Whilst ground truth labels could be used when verifying the results with the help of a synthetic dataset, validating the results using the collected data is much harder.

Qualitative validation usually happens through inspection or expert opinion. Meaning that a look is taken at the obtained clustering and it is argued if the results represent what was expected, or if there are obvious signs of misclustering in the final result. Although this is useful to obtain an initial indication about the accuracy of the results, no actual conclusion can be made.

Accompanying qualitative validation is usually quantitative validation. In this case results are validated through the use of evaluation metrics. However, each of these evaluation metrics have to be internal evaluation metrics meaning it is based on the clustering result and the original data and no ground truth is present. These internal validation metrics are all derived based off assumptions on what a good clustering is, which is not applicable for all types of data and might therefore give a false affirmation of the obtained results. Furthermore, clustering can have multiple solutions that all result in similar results for the evaluation metric making it still ambiguous to simply decide if a clustering is good based on quantitative validation.

1-3 Problem Statement

Imaging Mass Spectrometry is a technique that has many applications from biomedical exploration of organic tissue to use in forensic fingerprint analyses. To aid in the interpretability of the data, unsupervised learning is a powerful tool that can provide insight without prior information. Although rather vague at the current stage of the report, one of the unsupervised learning methods that is of interest is spectral clustering.

Spectral clustering is a graph based clustering approach known in the fields of image segmentation [32], community detection [33], speech separation [34] and text mining [35]. Most of these fields deal with high dimensional data and use spectral clustering for its most prominent feature, the ability to provide non-convex clusters.

In addition to this, a keen interest is shown towards spectral clustering due to two recently published papers. The first paper by Löffler et al. highlights that spectral clustering is optimal for Gaussian Mixture Models (GMM) [36]. The second paper by Delacour et al. has shown that IMS data can be modeled as a Spiked Mixture Model (SMM) [37]. Since a SMM is a constrained version of a GMM, we have reason to believe that similar guarantees will hold for the SMM. Although, at this point spectral clustering might seem like an alien subject, these

points will be further elaborated on throughout the report. The reason spectral clustering is introduced is to make the reader aware of the following research objective:

Research Objective

To develop and validate a spectral clustering framework for Imaging Mass Spectrometry (IMS) data that addresses computational scalability and resistance against noise, whilst maintaining biological segmentation.

Chapter 2

Theoretical Foundation

Spectral clustering is an unsupervised learning technique that belongs to the family of clustering algorithms. In this chapter, the problem formulation for spectral clustering is elaborated upon in section 2-1, followed by a explanation of the algorithm that will be used in section 2-2. While effective for non-convex partitions, spectral clustering faces scalability issues in IMS. Therefore, a memory-efficient approach is proposed to address computational constraints when handling large-scale datasets in section 2-4.

2-1 The Problem

Spectral clustering is a graph-based approach to clustering that has been widely used across various fields including image segmentation [32], community detection [33], speech separation [34] and text mining [35]. Its proven ability with high dimensional data, makes it particularly interesting for IMS datasets. To illustrate the problem spectral clustering addresses, we consider the example of bi-partitioning, which helps build intuition for why solving the problem in its original formulation is difficult. Spectral clustering avoids this difficulty by relaxing the original NP-hard problem, allowing for approximate but more efficient solution using the eigendecomposition of the graph Laplacian. T

Given a dataset $X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^d$, where each x_i is a pixel with a mass spectrum of dimension d, the goal is to group similar points into clusters while ensuring dissimilarity between different clusters. To achieve this, we start with a similarity measure (i.e. the gaussian similarity $s_{ij} = \exp(-||x_i - x_j||^2/2\sigma^2)$) using pairwise relationships between data points, with the metric being maximal when points are identical (i.e., when $x_i = x_j$). This leads to the construction of a similarity matrix $S = (s_{ij} \geq 0)_{i,j=1,\ldots,n} \in \mathbb{R}^{n \times n}$. This matrix is symmetric, with all entries non-negative and typically has a diagonal with all zeros ($s_{ii} = 0$) to avoid self-loops in the graph representation [38, 39].

The similarity graph can be constructed using a multitude of different approaches such as η-neighborhood graphs (connecting points within a distance threshold) or k-nearest neighbor graphs (linking each point to its k closest neighbors). Our focus will be on constructing

fully connected graphs, where every pair of data points is linked. Through this construction, global and local information is contained in the similarity graph, which in turn results in more information when clustering [38].

From the similarity matrix S, an undirected weighted graph $G = (V, E, W)$, where the set of vertices $V = v_1, v_2, \ldots, v_n$ correspond to the data points in X, the set of edges E connect pairs of vertices for which $s_{ij} > 0$, and values of the weight adjacency matrix $W = (w_{ij})_{i,j=1,\ldots,n}$ are set to the similarity values ($w_{ij} = s_{ij}$).

Given a constructed graph G the question being posed is [39]:

Minimum cut problem formulation

How can we obtain a separation of the graph into sets (A, A^c), whilst minimizing the total weight of edges being cut?

The cut weight is defined as the sum of the weights of edges crossing from A to $A^c = V \backslash A$, see Equation 2-1.

$$\text{cut}(A) = \sum_{i \in A, j \in A^c} w_{ij} \tag{2-1}$$

As there are no constraints imposed on the size of the set, minimizing the cut(A) leads to an optimal solution of $A = \emptyset$ with cut $(\emptyset) = 0$, which is a trivial partition. Before looking at solution of spectral clustering, let us examine a naïve approach to solving this problem, given that the trivial solution (the empty or full set) are not allowed [39].

Naïve approach Let us assume that our constructed graph has $n = 10$ vertices. This naïve approach addresses the problem combinatorially, which involves determining all possible ways to partition the vertices into two non-empty subsets A and A^c, counting the total weight of the cut edges for each partition and selecting the one with the lowest value.

For a graph with 10 vertices, the total number of possible partitions is $2^{n-1} - 1 = 2^9 - 1 = 511$, where the empty set and complete set are omitted, and it is accounted for the fact that $(A, A^c) = (A^c, A)$.

Although it is still feasible to compute for $n = 10$, the complexity of $\mathcal{O}(2^n)$ makes it impractical for IMS, where $n \sim 10^3 - 10^5$. Furthermore, although trivial solutions such as the empty or full set where not allowed, partitions where one set contains a singular data point are still possible. Thus, this would still result in uninformative partitions, highlighting the need for further restrictions in attainable partitions.

2-1-1 Graph Laplacian

Before we continue to the problem statement that is being solved with spectral clustering, a final component has to be introduced. The graph Laplacian (L) is defined as shown in Equation 2-2, where $D \in \mathbb{R}^{n \times n}$ is the diagonal degree matrix and its entries can be calculated by summing the rows of the weight matrix (Equation 2-3).

$$L = D - W \tag{2-2}$$

$$D_{ii} = \sum_j w_{ij} \tag{2-3}$$

The graph Laplacian is used to represent the connectivity of the data and is useful due to several properties shown below [38]:

Properties of the Graph Laplacian Matrix (L)

1. For every vector $f \in \mathbb{R}^n$ we have
$$f^T L f = \sum_{i<j}^{n} w_{ij}(f_i - f_j)^2. \tag{2-4}$$
2. L is symmetric and positive semi-definite
3. The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector $\mathbb{1}$.

Explanation of the properties The second and the third property are a consequence of the construction of the graph Laplacian. However, the first property needs more explanation, which allows the cut(A) to be presented in the quadratic form of the graph Laplacian.

Given a balanced bi-partition into sets A and A^c, an indicator vector $f \in \mathbb{R}^n$ can be defined where $f_i = 1$ if vertex $v_i \in A$ and $f_i = -1$ if vertex $v_i \in A^c$. The quadratic form of the graph Laplacian can be rewritten as follows:

$$f^T L f = f^T D f - f^T W f$$

The different terms can also be written as:

- The degree term $f^T D f = \sum_i d_i f_i^2 = \sum_i d_i = 2\sum_{i,j} w_{ij}$, since $f_i^2 = 1 \,\forall\, i$.
- The weight adjacency term $f^T W f = \sum_{i,j} w_{ij} f_i f_j$, where the product $f_i f_j = 1$ if the vertices are in the same set and $f_i f_j = -1$ if they vertices are in different sets.

Expanding the target expression in the first property results in the following expression:

$$\sum_{i<j}^{n} w_{ij}(f_i - f_j)^2 = \frac{1}{2}\sum_{i,j}^{n} w_{ij}(f_i - f_j)^2 = \frac{1}{2}\sum_{i,j}^{n} w_{ij}(f_i^2 + f_j^2 - 2f_i f_j) = \frac{1}{2}\sum_{i,j}^{n} w_{ij} f_i^2 + \frac{1}{2}\sum_{i,j}^{n} w_{ij} f_j^2 - \sum_{i,j} w_{ij} f_i f_j$$

Each of these terms can be expressed in terms of the aforementioned quadratic expression using D and W:

- The first term: $\frac{1}{2}\sum_{i,j}^{n} w_{ij} f_i^2 = \frac{1}{2}\sum_i^n f_i^2 (\sum_j^n w_{ij}) = \frac{1}{2}\sum_i^n f_i^2 d_i = \frac{1}{2} f^T D f$
- The second term can be derived similarly to the first term and also results in $\frac{1}{2} f^T D f$.

- The third term can be rewritten into $\sum_{i,j}^{n} w_{ij} f_i f_j = f^T W f$

When all terms are summed together it becomes apparent that property 1 holds:

$$\sum_{i<j}^{n} w_{ij}(f_i - f_j)^2 = \frac{1}{2} f^T D f \frac{1}{2} f^T D f - f^T W f = f^T D f - f^T W f = f^T L f$$

Example: Cut(A) and the Quadratic Form of the Graph Laplacian A short example will now follow to illustrate the concepts introduced so far. We will consider the graph displayed in Figure 2-1, this graph has 4 vertices and is fully connected with cross edge weights of 0.5 and weights of 1 for the edges that connect the nodes we would like to keep in our sets.

Figure 2-1: Example graph used to illustrate the introduced concepts. [Own Work]

It can already be seen that the optimal solution is cutting the cross edges which would lead to a cut value of 2. We will now verify that the quadratic form of the graph Laplacian provides the same result. To this end we will set up the W and D matrices:

$$W = \begin{pmatrix} 0 & 1 & 0.5 & 0.5 \\ 1 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 1 \\ 0.5 & 0.5 & 1 & 0 \end{pmatrix}, D = \text{diag}(2,2,2,2) \text{ giving } L = \begin{pmatrix} 2 & -1 & -0.5 & -0.5 \\ -1 & 2 & -0.5 & -0.5 \\ -0.5 & -0.5 & 2 & -1 \\ -0.5 & -0.5 & -1 & 2 \end{pmatrix}$$

The indicator vector f needs to be of the form $f = [1,1,-1,-1]^T$, if $A = \{1,2\}$ and $A^c = \{3,4\}$, which would be the optimal solution. This results a value for $f^T L f = 8$, it is clear that this is not the expected value of 2 for the cut(A). This difference can be explained by looking at the expanded expression of the quadratic form of the graph Laplacian, where contributions accumulate only for cut edges, resulting in $(f_i - f_j)^2 = (1 - (-1))^2 = 4$. Accounting for a correction factor leads to the corrected expression shown in Equation 2-5, and thus the cut(A) expressed as a quadratic form of the graph Laplacian also results in a value of 2 (8/4) matching the expected cut size.

$$\text{cut}(A) = \frac{1}{4} f^T L f = \frac{1}{4} \sum_{i<j}^{n} w_{ij}(f_i - f_j)^2 \tag{2-5}$$

Normalized Graph Laplacian Often the graph Laplacian is normalized before employing spectral clustering, to prevent bias towards nodes with higher degrees. There are two ways to construct a normalized graph Laplacian and both are closely related to each other [38]:

$$\mathcal{L}_{sym} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

$$L_{rw} := D^{-1} L = I - D^{-1} W$$

Where, $\mathcal{L}_{sym}$ stands for symmetrical graph Laplacian because the matrix is symmetric and L_{rw} stands for random walk graph Laplacian. However, for the purposes of this literature review $\mathcal{L}_{sym}$ is only used. As was the case with the unnormalized graph Laplacian, the normalized graph Laplacian also has interesting properties [38].

Properties of the Normalized Symmetric Graph Laplacian Matrix ($\mathcal{L}_{sym}$)

1. For every vector $f \in \mathbb{R}^n$ we have

$$f^T \mathcal{L}_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 . \tag{2-6}$$

2. $\mathcal{L}_{sym}$ is symmetric and positive semi-definite
3. The smallest eigenvalue of L is 0, the corresponding eigenvector is $D^{1/2}\mathbb{1}$.

2-1-2 Spectral Clustering

Balanced Partitions Let us revisit the minimum cut problem. So far, we have defined a notion of a cut that helps us move toward the problem solved by spectral clustering. However, with the current formulation, there is still the possibility of obtaining a trivial solution, such as the empty set. In our earlier example, we introduced the indicator vector f. By minimizing $\text{cut}(A)$ subject to the constraint that $f \in \{\pm 1\}^n$, where $f_i = 1$ if $i \in A$ and $f_i = -1$ if $i \in A^c$, and additionally enforcing the balance condition $\mathbb{1}^T f = 0$, which ensures the partition is balanced, such trivial solutions are avoided.

$$\begin{aligned} \min \ & \frac{1}{4} f^T L f \\ \text{s.t. } & f \in \{\pm 1\}^n \\ & \mathbb{1}^T f = 0 \end{aligned}$$

Unfortunately, this problem is too constrained. First, to satisfy $\mathbb{1}^T f = 0$ this problem formulation requires that $|A| = |A^c|$ meaning that there can not be an uneven amount of data points. Furthermore, this problem still needs us to check all possible ways $f \in \{\pm 1\}^n$ can be constructed making it impractical.

A method to evaluate if an obtained partition is not too small , whilst removing the possibility of trivial solutions is called the Cheeger's cut.

Cheeger's Cut and Cheeger's Constant [ ]

Given a graph and a vertex partition (A, A^c), the cheeger cut (also known as conductance, and sometimes expansion) of A is given by

$$h(A) = \frac{\text{cut(A)}}{\min\{\text{vol(A)}, \text{vol(A}^c)\}}, \tag{2-7}$$

where $\text{vol(S)} = \sum_{i \in S} deg(i)$. Also, the Cheeger's constant of G is given by

$$h_G = \min_{A \subset V} h(A) \tag{2-8}$$

The Cheeger cut ($h(S)$) is a measure that quantifies the ratio of crossing edges between sets to the volume for any subset of vertices. The minimum value of $h(S)$ is called the Cheeger constant (h_G) and represents the optimal graph partition. However, this problem suffers from the same issue as before, the minimum value is calculated over all possible subsets $A \subset V$ and as there are $2^{|V|}$ possible subsets it is computationally intractable for many cases. This makes it a NP-hard problem, meaning that a solution can not be found in polynomial time.

The reason the Cheeger cut and constant are introduced is because the attainable partitions of the upcoming problem formulations can be bounded using the Cheeger constant.

Normalized Cut A way to relax the balanced partition problem is by allowing $f \in \{a, b\}$, where both a and b are real distinct values [39, 38]. The balancing constrained can than be relaxed by requiring that,

$$a \text{ vol}(A) + b \text{ vol}(A^c) = 0,$$

which allows A and A^c to be of different sizes. This constraint can also be reformulated into $\mathbb{1}^T Df$, as the volume is nothing but the sum of degrees. Additionally, some sort of normalization for a and b can be determined by requiring that

$$a^2 \text{ vol}(A) + b^2 \text{ vol}(A^c) = 1,$$

which can be rewritten to $f^T Df = 1$. This constraint ensures that no trivial solution are obtained. Lastly, the objective function needs to be changed to correspond to the so-called normalized cut ($\text{Ncut}(A)$). The derivation will be omitted as it does not bring any additional information, but the $\text{Ncut}(A)$ is equal to the following,

$$\text{Ncut(A)} = \text{cut}(A) \left[\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(A^c)} \right] = f^T Lf,$$

which is the quadratic form of the graph Laplacian. The complete problem can be described as,

$$\begin{aligned} &\min f^T L f \\ &\text{s.t. } f \in \{a, b\}^n \text{ for some a,b} \\ &\quad y^T D y = 1 \\ &\quad y^T D \mathbb{1} = 0. \end{aligned}$$

This problem remains NP-hard, but is crucial for the final relaxation. The Ncut(A) can be related to the Cheeger cut, by looking at the formulation of both problems and noticing that,

$$h(A) \leq \text{Ncut}(A) \leq 2h(A).$$

Eigenvector Problem For the final formulation of the problem, the Ncut(A) problem will be relaxed in two ways. First, the constraint that f can only take on values a and b will be changed to $f \in \mathbb{R}^n$ [38, 39]. This turns the problem from a discrete partitioning problem to a continuous one. Secondly, the graph Laplacian will be normalized to clearly show that the resulting problem will become a eigenvector problem and therefore computationally tractable. Spectral in spectral clustering refers to the use of this eigen decomposition to solve the optimization problem. This also means that instead of f, a transformed variable $z = D^{1/2} f$ is introduced to maintain the objective function in the same form.

$$\begin{aligned} &\min z^T \mathcal{L}_{sym} z \\ &\text{s.t. } z \in \mathbb{R}^n \\ &\quad \|z\|^2 = 1 \\ &\quad (D^{1/2}\mathbb{1})^T z = 0 \end{aligned} \tag{2-9}$$

The minimum for 2-9 is obtained by the second smallest eigenvalue of the normalized laplacian $\lambda_2(\mathcal{L}_{sym})$, as $\lambda_1(\mathcal{L}_{sym}) = 0$ and its accompanying eigenvector is $(D^{1/2}\mathbb{1})^T$. Furthermore, all eigenvalue are ordered from smallest to largest $(0 = \lambda_1(\mathcal{L}_{sym}), \ldots, \lambda_n(\mathcal{L}_{sym}))$. The second eigenvector, in the form of z, would be orthogonal to the first eigenvector and thus satisfy the constraints.

To obtain the subsets (A, A^c), a threshold τ can be set and $A = \{i \in: z \leq \tau\}$. As z has at most a size of n, one could try all possible values of τ and select the best partition, which would have a complexity of $O(n)$.

The obtained solution now is continuous and therefore it can be concluded that,

$$\lambda_2(\mathcal{L}_{sym}) \leq \min_{A \subset V} \text{Ncut}(A)$$

This also implies that

$$\lambda_2(\mathcal{L}_{sym}) \leq h_G$$

which provides a lower bound on the Cheeger constant, and allows us to determine bounds on the partitions obtained using the problem in 2-9.

The derivation of the upper bound will be omitted due to complexity, but the Cheeger constant is bounded by

$$\lambda_2(\mathcal{L}_{sym}) \leq h_G \leq \sqrt{2\lambda_2(\mathcal{L}_{sym})}.$$

Let our obtained partition with 2-9 have a Cheeger cut of $\hat{h}(A)$, than this can by realizing that

$$h_G \leq h(A) \leq \sqrt{2\lambda_2(\mathcal{L}_{sym})}.$$

Using the lowerbound, to rewrite the upperbound in terms of h_G results in the Cheeger cut ($\hat{h}(A)$) of our obtained partition to be at most a factor of $2\sqrt{h_G}$ from the optimal partition given by the Cheeger constant.

$$h_G \leq \hat{h}(A) \leq 2\sqrt{h_G}$$

Extension to multiple clusters An extension of the normalized cut can be made for more than 2 clusters and a derivation can be found in the paper titled "A Tutorial on Spectral Clustering" by Von Luxberg [38]. This problem can than again be relaxed to allow for real values and be transformed into an eigenvector problem. The problem has now become too complex to solve by thresholding the second eigenvector, the algorithm used to solve this problem will be discussed in section 2-2. The main focus of this section is to show that the k-way expansion of the problem still has bounds on the partition attained by the algorithm.

First, we will have to introduce the k-way Cheeger's cut as we are now considering $k > 2$:

k-way Cheeger's cut and Cheeger's Constant [ ]

For any subset $A \subseteq V$, the Cheeger's cut of A is given by the aforementioned $h(A)$ of Equation 2-7. Subsequently, we will define for every $k \in \mathbb{N}$ the k-way Cheeger's constant:

$$h_G(k) = \min_{A_1, A_2, \cdots, A_k} \max\{h(A_i) : i = 1, 2, \cdots, k\} \tag{2-10}$$

where the minimum is taken over the k non-empty disjoint subsets $A_1, A_2, \cdots, A_k \subseteq V$.

Here, we want to find a k-way partition where the worst subset (in terms of Cheeger cut) has the smallest possible Cheeger cut among all possible partitions.

The Cheeger inequality for any graph G and cluster size $k \in \mathbb{N}$, is than formulated in Equation 2-11 [41, 40].

$$\frac{1}{2}\lambda_k(\mathcal{L}_{sym}) \leq h_G(k) \leq Ck^2\sqrt{\lambda_k(\mathcal{L}_{sym})} \tag{2-11}$$

Here, $C > 0$ is a universal constant. It can be seen that the lower bound has remained in a similar form to the case of $k = 2$, whilst the upperbound has become looser. The lower bound has been derived analytically in similar way to that of case of $k = 2$, whilst the upperbound has been proven algorithmically.

Similarly to the case of $k = 2$, we can rewrite this inequality into the following form to bound the obtained partition $h(S_i)$:

$$h_G(k) \leq \hat{h}(S_i) \leq Ck^2\sqrt{2h_G(k)} \tag{2-12}$$

Evidently this bound is less tight than the case of $k = 2$, as it scales with number of clusters.

2-2 Algorithm

In subsection 2-1-2 an algorithm was explained that is able to solve the bi-partition ($k = 2$) problem for the eigenvector problem. Depending on the application, this is often insufficient and $k > 2$ is chosen. An algorithm that accounts for that is the one by Ng et al. shown in algorithm 1 [42].

Spectral clustering with algorithm 1 uses the first k eigenvectors of the Laplacian matrix L to embed data into a lower-dimensional space, where k-means clustering is applied. The reason for choosing only the first k eigenvectors can be explained in two fold.

The first interpretation uses the quadratic form of the graph Laplacian, which is used in the objective function of all problem formulation shown in section 2-1. This form allows us to find the path that minimizes the amount of edges being cut to partition the graph, as the cut(A) and Ncut(A) can both be expressed in the quadratic form of the laplacian. Therefore, selecting too many eigenvectors will only dilute the signal for the most informative directions and could lead to the introduction of noise. Furthermore, to identify k-clusters, a k-dimensional embedding is required. Therefore, selecting too many clusters can be detrimental as this could lead to noisy clustering results.

Another way of looking at it is through the the spectral gap. The spectral gap refers to a large difference between eigenvalue λ_k and λ_{k+1}, this large cap serves as indication for k well separated clusters [38, 43, 44].

As an intuitive example, one could think about a making a graph for k well separated clusters. The similarity graph would have strong connections within the clusters and weak connections between the clusters. This means that the weight matrix would exhibit a block diagonal structure, where each block would be a distinct cluster. This structure would be transferred over to L, such that the first k eigenvectors would highlight nodes in the k-clusters, whilst the $k + 1$ eigenvector would mostly be referring to noise.

If we assume that the matrix of L is exactly block diagonal, than each of the first k- eigenvectors would contain values in $\mathbb{R}$ at the entries that correspond to the size of the block and 0's for the other ones, see Figure 2-2. This makes sense as each of the blocks on the diagonal are Laplacian matrices, each having the $\mathbb{1}$ as the eigenvector associated with the smallest eigenvalue of 0 [38]. Therefore, such a matrix would contain as many 0 eigenvalues as blocks.

The $k+1^{th}$ eigenvector would not contain such a division and its eigenvalue would be greater than 0.

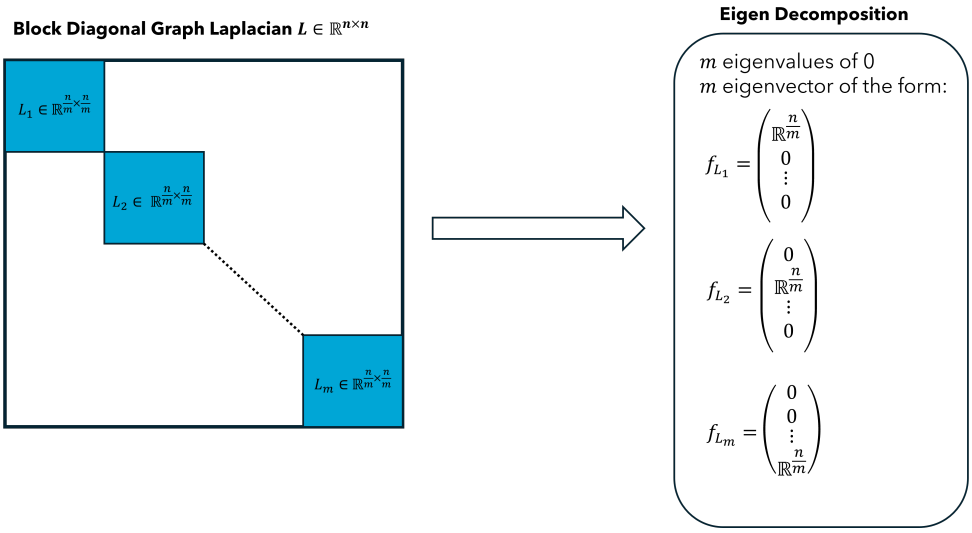


Figure 2-2: Visualization of the eigendecomposition for blockdiagonal graph Laplacians.[Own Work]

Although the spectral gap is a useful measure to get an indication about the amount of clusters that one should select, it deteriorates in performance when data has noise and overlap as can be seen in Figure 2-3. Therefore, one should mostly use this metric to get an initial idea about the amount of clusters that can be used.

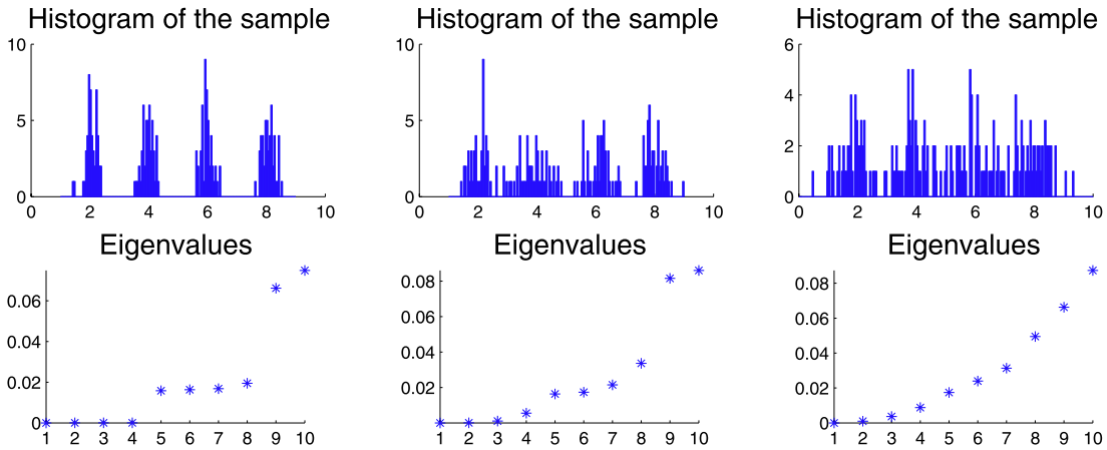


Figure 2-3: Spectral gap vanishing with increasing overlap and noise between the clusters. [Image from [38]]

Algorithm 1 Normalized Spectral Clustering (Ng et al., 2002)

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number of clusters k.

1: Construct a similarity graph. Let W be its weighted adjacency matrix.
2: Compute the normalized Laplacian $\mathcal{L}_{\text{sym}}$:

$$\mathcal{L}_{\text{sym}} = I - D^{-1/2} W D^{-1/2},$$

where D is the degree matrix.
3: Compute the first k eigenvectors $u_1, u_2, \ldots, u_k$ of $\mathcal{L}_{\text{sym}}$.
4: Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the eigenvectors $u_1, u_2, \ldots, u_k$ as columns.
5: Form the matrix $T \in \mathbb{R}^{n \times k}$ by normalizing each row of U to have unit length:

$$t_{ij} = \frac{u_{ij}}{\sqrt{\sum_{j=1}^{k} u_{ij}^2}}.$$

6: For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of T.
7: Cluster the points $\{y_i\}_{i=1}^{n}$ with the k-means algorithm into clusters $C_1, C_2, \ldots, C_k$.

Output: Clusters $A_1, A_2, \ldots, A_k$, with:

$$A_i = \{j \mid y_j \in C_i\}.$$

2-3 Strengths and Limitations of Spectral Clustering

So far, the advantages of spectral clustering have not been mentioned and the focus was on understanding the theoretical foundation behind spectral clustering. Therefore, this section will focus on obtaining a better understanding of the advantages and limitations spectral clustering brings.

Strength: Arbitrarily Shaped Clusters The graph-based approach in spectral clustering enables the discovery of clusters with arbitrary shapes. Once the graph Laplacian is constructed, partitions are obtained by minimizing the total cost of cut edges, with no assumptions on the cluster shape. This stands in contrast to traditional methods like k-means and Gaussian mixture model (GMM) clustering.

In k-means, data points are assigned to clusters based on their distance to a centroid, typically using metrics such as euclidean or cosine distance. This results in partitions that favor spherical or convex shapes, making it challenging to identify arbitrarily shaped or non-convex clusters.

GMM clustering allows for slightly more flexibility by modeling the data as a mixture of gaussian distributions, allowing for elliptical shapes through covariance matrices. However, even though GMMs can approximate more complex cluster shapes through the mixtures of gaussians, they are still limited in capturing highly non-convex or irregular clusters, as each cluster remains convex.

This property of spectral clustering could lead to better segmentation of the data, as different clustering results will be obtained by these methods if the data indeed contains regions that can only be captured through methods that can cluster non-convex regions.

Strength: Cheeger Inequality Bounds Although spectral clustering does not guarantee the optimal partition due to its relaxation of the NP-hard problem, the Cheeger inequality provides valuable bounds on the quality of the approximated clusters. This gives a quantitative measure of the clustering's effectiveness, with tighter bounds indicating better alignment to the optimal solution.

Strength: Speculated Optimality of Spectral Clustering on IMS Data Löffler et al. have shown that spectral clustering is optimal for GMM's [36]. Furthermore, recent results obtained by Delacour et al., have shown that IMS data can be approximated by a Spiked Mixture Model (SMM) [37]. A SMM is a constrained version of a GMM, therefore it would not be a stretch to assume that similar guarantees could be derived for the case of SMM.

Limitation: Time and Memory Complexity The biggest limitation of spectral clustering is with regards to the time and memory complexity of the algorithm. Algorithm 1 can be broken down into the following steps:

1. Constructing the similarity matrix.
2. Computing the normalized Laplacian
3. Computing the eigendecomposition on the normalized laplacian.
4. Applying k-means clustering on the eigenvectors.

For the time complexity, the computation of the similarity matrix has a time complexity of $\mathcal{O}(n^2 d)$, where d is added to account for high dimensional data. The quadratic term comes from evaluating similarities for every pair of points.

The computation of the normalized laplacian has a time complexity of $O(n^2)$, due to matrix operations on the $n \times n$ similarity matrix.

The eigen decomposition has the highest time complexity of $O(n^3)$. However, as only the k-smallest eigenvectors and eigenvalues are required, this can be reduced to $O(n^2 k)$.

In contrast, an algorithm that is used to its efficiency, k-means has a time complexity of $O(nkdi)$, which in the case of IMS data is mostly dominated by n. Furthermore, algorithms such as k-means++, use improved initialization strategies which would reduce the number of iterations.

This makes the dominant term $O(n^2 d)$, because $d >> k$ in case of IMS data. However, depending on the application it could be interchanged with $O(n^2 k)$.

The quadratic time complexity in n is not a favorable feature of spectral clustering, as this slows down computation significantly compared to methods that are linear in n such as k-means.

For the memory complexity, a similar analysis can be performed. The eigenvectors, degree matrix and k-means input all have a complexity that is linear in n (i.e. $O(n)$). Whilst the normalized graph Laplacian and the similarity matrix require matrices of size $n \times n$ to be stored, and therefore have a complexity of $O(n^2)$.

This memory constraint is especially a big limitation when analyzing large datasets. Typical work laptops or computers have a memory of 16 to 32 GB nowadays. However, if a dataset has $n = 10^5$ points this requires approximately ~ 75 GB of memory to be stored in float64 format. This clearly illustrates an issue in scalability for spectral clustering, which in the context of IMS is an issue.

Limitation: Sensitivity to Choice of Similarity Metric The choice of similarity metric impacts the obtained clustering results in spectral clustering greatly. The similarity matrix is a key component in constructing the graph Laplacian and defines the pairwise relationship between data points, affecting the obtained clusters.

For example, if a gaussian similarity metric is selected the choice of σ impacts how much importance you give to local similarity. A higher σ means that you care about a broader region, which can lead to over-smoothing the similarity values. Whilst a smaller σ constraints it more, but a σ that is too low can lead to fragmented clusters.

On the other hand, the cosine similarity does not use the euclidean distance but focuses on directional alignment through the cosine angle. This approach normalizes the magnitude and relies on orientation, often yielding different similarity matrices.

$$\text{Gaussian Similarity Metric} = \exp\left(\frac{||x_i - x_j||^2}{2\sigma^2}\right)$$

$$\text{Cosine Similarity Metric} = 2 - \frac{x_i \cdot x_j}{||x_i||_2 ||x_j||_2}$$

2-4 Memory Constrained Spectral Clustering

From the limitations of spectral clustering, it is apparent that scalability emerges as a critical problem for large datasets. In this section a heuristic is proposed that allows one to implement spectral clustering on large IMS datasets.

The Algorithm Given our dataset $X \in \mathbb{R}^{n \times d}$, the data set will be split into r distinct subsets. These r distinct subsets are selected through random seeding, meaning a random data point ($x_{r,i}$, the r is used to indicate which subset it belongs to) is selected and the subset is constructed by looking at the $[n/r] - 1$ data points that have the highest similarity to $x_{r,i}$. The variable r should be chosen as small as admissible by the hardware. This should be done to minimize the loss of information between data points, the larger r is the smaller the subsets, which can lead to fragmented clusters.

Since the subsets are not overlapping, spectral clustering can be applied independently to each subset.The final clustered image can be constructed by collecting the labels obtained for

each subset, see algorithm 2. This method reduces the memory complexity from $O(n^2)$ to $O((n/r)^2)$, which is still quadratic in n but implementable within hardware limitations.

For each subset r, construct a fully connected weight adjacency matrix W_r. From this, derive the individual graph Laplacian $L_r = D_r - W_r$ and normalized Laplacian $\mathcal{L}_r = D_r^{-1/2} L_r D_r^{-1/2}$. Collecting all W_r into a global weight adjacency matrix W yields a block-diagonal structure, as intersubset similarities are ignored. Consequently, the global Laplacian L and normalized Laplacian $\mathcal{L}_{sym}$ also remain block-diagonal, since D is diagonal and W follows the same pattern. This resembles the decomposition shown in Figure 2-2.

Addressing Limitations There are some clear drawbacks from this method, the most apparent one being the divisibility of n/r. In case a remainder exists after division, one can choose to assign the remainder to a specific subset which will be negligibly larger.

Another limitation is that the subsets are constructed with respect to a random seeding point. This could lead to data points of a region of interest being disconnected into different subsets. As an example, given $n = 30,000$ and $r = 2$, this means that each subset will contain 15,000 data points. Now imagine that the similarity with respect to the random seeded point is high initially but degrades as more data points are accumulated into the subset. This could mean that the tail end of the n/r most similar data points have a low similarity value with respect to the random seeding point, but could have a high similarity with respect to data points in the second subset.

Finally, the amount of clusters that are selected will be constant for all subsets. This means that the total amount of clusters are equal to kr, which is a limitation as certain subsets might require less clusters than others. Variable cluster allocation is difficult to achieve as unsupervised learning means that there can not be any prior information.

Algorithm 2 Memory-Constrained Spectral Clustering for Large IMS Datasets

Input: Dataset $X \in \mathbb{R}^{n \times d}$, number of subsets $r \in \mathbb{N}$, similarity metric $s(\cdot, \cdot)$, optionally: number of clusters $k \in \mathbb{N}$

Output: Assignment of clusters $A_1, A_2, \ldots, A_r$

1: **for** $i = 1$ **to** r **do**
2: Select a random seed data point $x_{i,\text{seed}}$ from the remaining pool.
3: Compute similarities $s(x_{i,\text{seed}}, x_j)$ for all remaining x_j in X.
4: Form subset X_i by selecting $n/r - 1$ data points most similar to $x_{i,\text{seed}}$ (including the seed itself).
5: Remove selected points from the pool.
6: Compute the similarity matrix $S_i \in \mathbb{R}^{(n/r)\times(n/r)}$ for X_i using $s(\cdot, \cdot)$.
7: Apply spectral clustering (Algorithm 1) to S_i, obtain labels $y^{(i)}$.
8: **end for**
9: Concatenate labels from all subsets to obtain global cluster assignments.
10: **return** Cluster assignments $A_i = \{j \mid y_j \in C_i\}$ for all i.

Chapter 3

Data and Experiments

3-1 Synthetic Dataset

The use of a synthetic dataset facilitates the validation of spectral clustering through direct comparison with other clustering methods against a known ground truth. As the aim of this research assignment is to investigate the performance of spectral clustering data with high noise and compare it against k-means clustering in the context of IMS, the data is modelled to closely represent IMS data. The paper by Delacour et al. [37], which is used as motivation for the use of spectral clustering, has concluded that a SMM represents IMS data better than a GMM. Therefore, our dataset will be constructed using the SMM shown below as a guideline:

$$\mathbf{y} = \begin{cases} \alpha\mathbf{x}_1 + \varepsilon & \text{with probability } \pi_1, \\ \vdots & \\ \alpha\mathbf{x}_K + \varepsilon & \text{with probability } \pi_K, \end{cases} \tag{3-1}$$

$$\alpha \sim \mathcal{N}(0,1), \quad \varepsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I}),$$
$$\sum_{k=1}^{K} \pi_k = 1, \quad \mathbf{x}_1, \ldots, \mathbf{x}_K \in \mathbb{R}^d.$$

Where $\mathbf{y_1}, \cdots, \mathbf{y}_N \in \mathbb{R}^d$ are N independent observations sampled from the above model. The parameter α is a random scaling factor of observation $\mathbf{y}$, $\mathbf{x}_k$ is the k-th subpopulation or spike, ϵ is the random noise of observation $\mathbf{y}$, and π_k is again the mixing coefficient or as mentioned in the paper the probability of the k-th subpopulation.

$$p(\mathbf{y}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(0, \Sigma_k)$$

This model can not be used directly to create mass spectra, as the variable $\alpha \sim \mathcal{N}(0,1)$ can be negative, and a mass spectrum can not have negative intensities, see (Figure 3-1). For this reason, the alpha parameter is chosen uniformly between $\alpha \sim U(\alpha_{min}, \alpha_{max})$ to imitate varying intensities in the mass spectrum. The standard values for $[\alpha_{min}, \alpha_{max}] = [0, 1.5]$. The gaussian noise ($\epsilon$), will not be adjusted and is maintained to imitate the background noise, the value for σ will be variable to simulate different levels of noise.

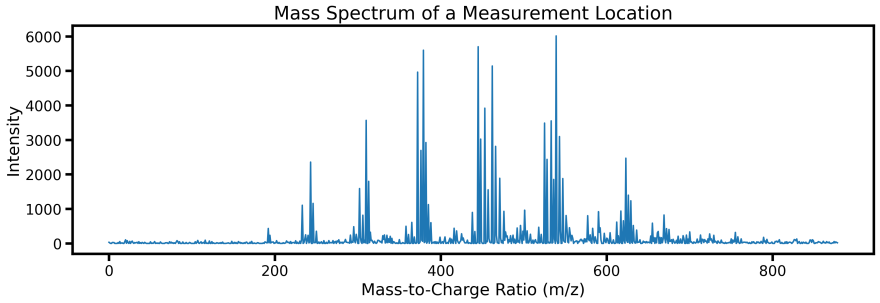


Figure 3-1: Example of a mass spectrum.

The synthetic dataset consists of four non-overlapping regions, each characterized by a distinct $\mathbf{x}_k$, with other observations generated according to Equation 3-1. The choice of one ground signal ($\mathbf{x}_k$) per region is due to deterministic nature of the clustering algorithms, where each pixel can only belong to one cluster. Although the underlying model of the generated data permits soft assignments, allowing a pixel to belong to more than one cluster via the mixing coefficients π_k, spectral clustering and k-means are a deterministic method that assign each pixel to exactly one cluster.

In Figure 3-2, the outlines of the different regions, the mask of the regions and the amount of data points are shown. The spikes corresponding to each region are shown in Figure 3-3 and some example observations generated using these spikes are shown in Figure 3-4.

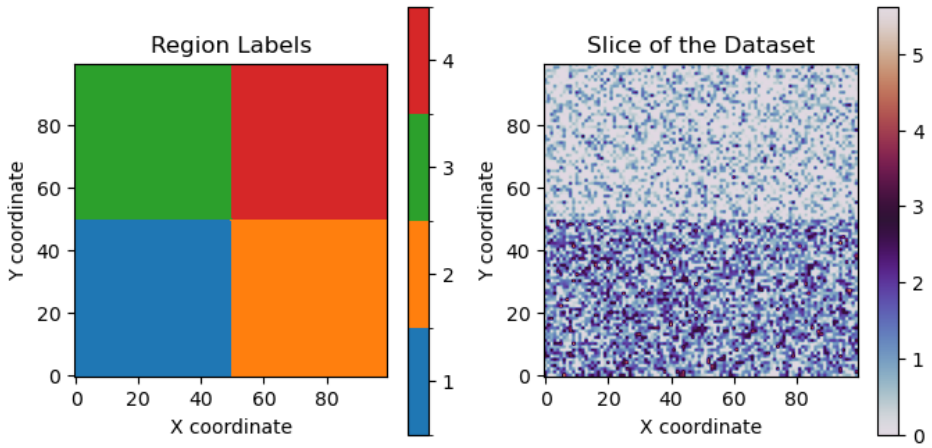


Figure 3-2: The different regions (or pixel locations) defined for generating the observations. [Own Work]

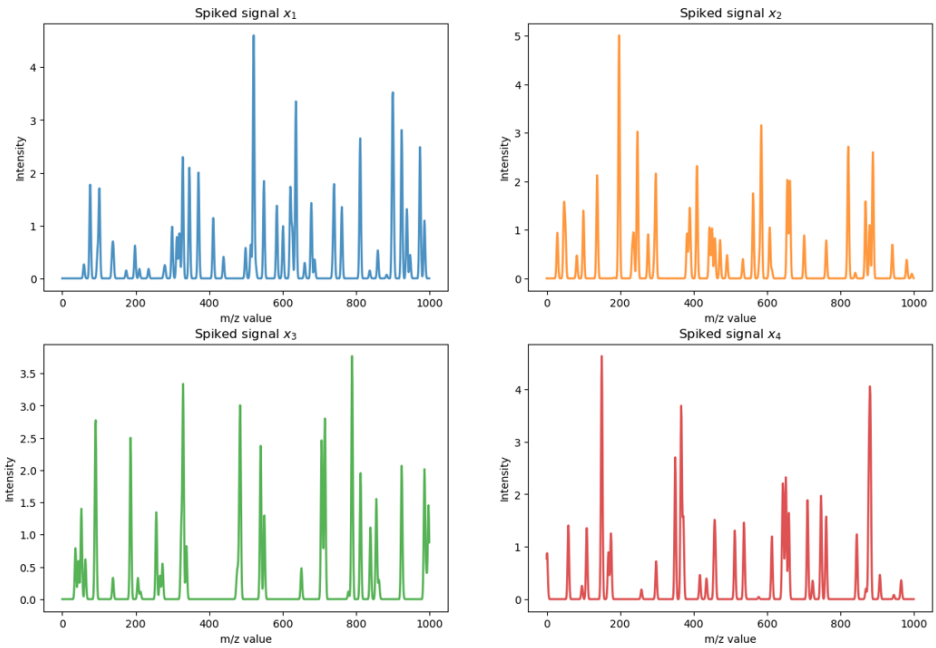


Figure 3-3: The true spikes ($\mathbf{x}_k$) of each region.

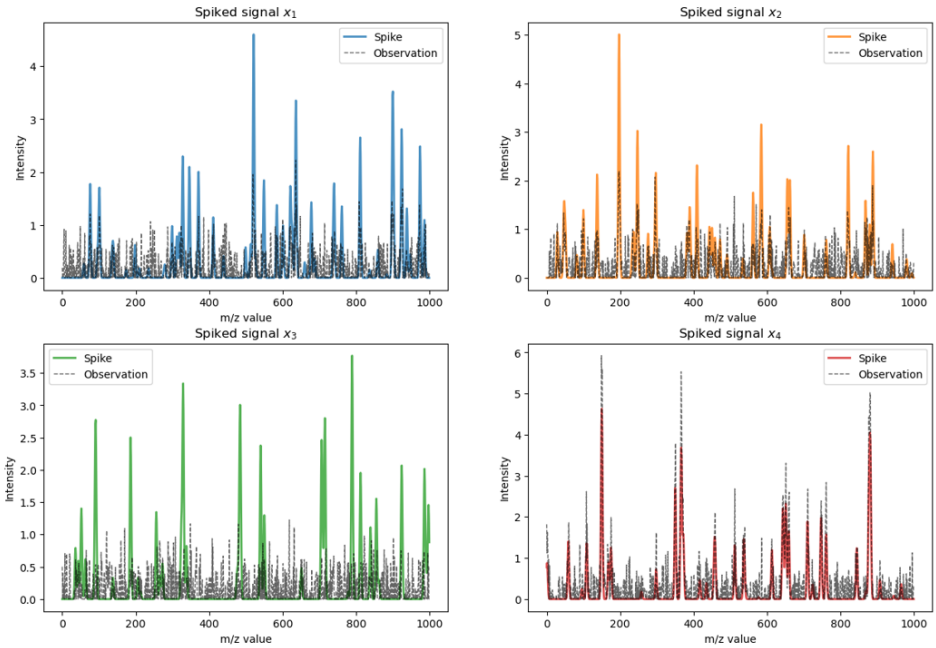


Figure 3-4: A subset of 4 observations, 1 from each region to visualize the observations and the impact of the intensity scaling parameter (α) and the gaussian noise (η) with a $\sigma = 0.4$.[Own Work]

3-1-1 Experiment 1: Performance of Spectral Clustering in the Presence of Noise

In this experiment, the performance of spectral clustering will be analysed on the aforementioned data set. The synthetic dataset will be generated for different values of gaussian noise controlled with the parameter σ. This parameter will be varied from between 0 and 2, where 0 would indicate clustering without noise and therefore only signals that are scaled using the parameter α. The amount of clusters will be assumed to be known and set equal to $k = 4$.

This will be done through comparison against a method that is used frequently for clustering on IMS data, which is called k-means clustering [18, 21, 6, 45]. As for spectral clustering the choice of similarity measure is of importance, two different metrics will be used, which are the euclidean metric and the cosine metric. A short discussion will now follow on the different metrics used, followed by a theoretical introduction into k-means clustering.

The metrics that will be used to quantify the results are the hamming loss (HL), adjusted rand index (ARI) and the confusion matrix. These indices all evaluate the obtain clustering results by comparing it with the ground truth.

Euclidean Metric The euclidean distance calculates the shortest straight line path between two data points, meaning that it uses the magnitude of data points. For k-means this means that the objective function in Equation 3-2, minimizes the distance of the data points with respect to a centroid.

The euclidean distance can not be directly used in spectral clustering, as here a similarity measure is needed. To transform the euclidean distance into a similarity measure, the inverse is taken and one is added to the denominator to prevent division by 0. This way a larger distance will lead to a lower similarity value, whilst a larger value will mean that points are closer together in terms of magnitude.

However, as mentioned in the section 1-2, distances tend to become uniform with growing dimensionality making, making data points that discrepancy between far and close redundant. As the current dataset contains $d = 879$ m/z bins, this should be of impact to the gaussian similarity metric and therefore the performance of spectral clustering.

$$\text{Euclidean Distance Metric} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$\text{Euclidean Similarity Metric} = \frac{1}{\text{Euclidean Distance} + 1}$$

Cosine Metric The cosine distance is defined as a shifted version of the cosine similarity. In contrast to euclidean metric, the cosine metric only looks at the orientation of the datapoints, it computes the angle between datapoints. The cosine similarity is defined as follows:

$$\text{Cosine Similarity} = \cos(\theta) = \frac{x_i \cdot x_j}{||x_i||_2||x_j||_2}$$

As one can see, the denominator normalizes the numerator, making the magnitude redundant. The cosine similarity measure in its current form can take on values in $[-1, 1]$, where the 1

means that the data points have the same direction high similarity and -1 that they are opposing each other. Similarity measures can not be negative as to not ruin the semi-positive definiteness of the graph Laplacian. Therefore, the cosine similarity is shifted with 1 to make the bounds between $[0, 2]$.

$$\text{Cosine Similarity Metric} = 1 + \cos(\theta) = 1 + \frac{x_i \cdot x_j}{||x_i||_2||x_j||_2}$$

The cosine distance is defined as,

$$\text{Cosine Distance Metric} = 1 - \text{Cosine Similarity} = 1 - \frac{x_i \cdot x_j}{||x_i||_2||x_j||_2}$$

The bounds of the cosine distance metric are between $[0, 2]$, where 0 indicates that data points are identical and 2 that they are completely dissimilar. The cosine similarity has been used in many high dimensional applications, such as text mining and natural language processing [46, 47], making it an interesting option for application in spectral clustering.

k-means Clustering The general formulation of k-means is displayed below [39, 28].

k-means problem statement

Given $X \in \mathbb{R}^{nxp}$, k-means clustering partitions the data points in to clusters $S_1 \cup \cdots \cup S_k$ with centers $\mu_1, \cdots, \mu_k \in \mathbb{R}^p$ as a solution to.

$$\min_{\substack{\text{partition} \\ S_1,\ldots,S_k \\ \mu_1,\ldots,\mu_k}} \sum_{l=1}^{k} \sum_{i \in S_l} \text{distance metric}(x_i, \mu_l). \tag{3-2}$$

The choice of distance metric depends on the application. In our case, the euclidean distance and the cosine distance are considered. Finding an exact solution to the aforementioned optimization problem is NP-hard, which necessitates the need for approximate solutions. A commonly utilized algorithm is Lloyd's algorithm; 1) Calculate/Determine the k cluster centroids, 2) Calculate the distance between each pixel and the cluster centroid, 3) Assign each data point to closest centroid, 4) Repeat until convergence criterion is met [28, 48]. This algorithm of k-means clustering aims to minimize the intra-cluster distance [48]. The existing implementations for k-means, in well reputed python packages, such as `Scikit Learn`, only work with an euclidean distance objective function. This is not surprising as it has been well studied and has convergence guarantees. An overview of Lloyds algorithm for k-means clustering is also presented in algorirthm 2 [48].

When utilizing the cosine distance, our attention will shift to a method called spherical k-means. Here, the data X will be L_2 normalized, such that $||x_i|| = 1$ for all data points, which means that the data has been projected on the unit hypersphere in $\mathbb{R}^d$ [49, 50]. The data has to be normalized to ensure the algorithm only considers the orientation of the data. Most of Lloyd's algorithm remains the same as for euclidean k-means, the main difference is the objective function and the centroid update step. Specifically, he centroid update step will use the Fisher mean,

$$\mu_i = \frac{\sum_{x_j \in S_i} x_j}{||\sum_{x_j \in S_i} x_j||}.$$

The use of the Fisher mean requires L_2 normalization in the data preprocessing step [49, 50]. To illustrate this, let consider a simple dataset $X = x_a, x_b$, where $||x_a|| >> ||x_b||$, than the Fisher mean would be biased towards x_a, eventhough the goal is to not rely on the magnitude of the data points. In algorithm 3, an overview is provided of the steps in spherical k-means [50].

Algorithm 3 k-means Clustering Algorithm (Lloyd, 1982).

Input: Dataset $X \in \mathbb{R}^{n \times d}$, number of clusters k.

1: Initialize k centroids $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^d$.
2: **repeat**
3: For each data point x_i, assign it to the closest centroid:

$$S_j = \{x_i \in X \mid ||x_i - \mu_j||_2^2 \leq ||x_i - \mu_l||_2^2 \quad \forall \quad 1 \leq l \leq k\}.$$

4: Update each centroid as the mean of its assigned points:

$$\mu_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i.$$

5: **until** Convergence criterion is satisfied.

Output: Clusters $S_1, S_2, \ldots, S_k$.

Confusion Matrix The confusion matrix illustrates which labels have been clustered correctly and which have been assigned to the wrong label. For a binary classification task, the confusion matrix is shown in Figure 3-5. It consists of four regions, each indicating how predicted labels align with the ground truth. A true positive occurs when the ground truth and predicted label match for the positive class, a false positive means the prediction assigns it to the positive class incorrectly, a true negative shows correct classification as negative, and a false negative indicates that negative classification was incorrectly marked as positive. For multi class tasks, the idea is similar, but the matrix is $k \times k$ in size for k clusters. Here, entries should be interpreted as points correctly clustered as x or mislabeled as x when they belonged to y.

Algorithm 4 Spherical k-means Clustering Algorithm (Dhillon and Modha, 2001).

Input: Dataset $X \in \mathbb{R}^{n \times d}$, number of clusters k.

1: Normalize all data points to unit length:

$$x_i = \frac{x_i}{||x_i||}, \quad \forall \quad 1 \leq i \leq n.$$

2: Initialize k centroids $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^p$ all L_2 normalized.
3: **repeat**
4: For each data point x_i, assign it to the centroid with highest cosine similarity:

$$S_j = \{x_i \in X \mid x_i^T \mu_j \geq x_i^T \mu_l \quad \forall \quad 1 \leq l \leq k\}.$$

5: Update each centroid using the Fisher mean:

$$\mu_j = \frac{\sum_{x_i \in S_j} x_i}{||\sum_{x_i \in S_j} x_i||} \quad \forall \quad 1 \leq j \leq k$$

6: **until** Convergence criterion is satisfied.

Output: Clusters $S_1, S_2, \ldots, S_k$.

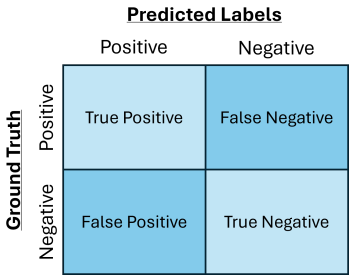


Figure 3-5: Confusion matrix for binary classification tasks.[Own Work]

Hamming Loss (HL) The hamming loss function compares the predicted labels against the ground, and sums the amount of misclassified labels (Equation 3-3).

$$\text{Hamming Loss} = \frac{1}{nL} \sum^{n_i} = 1 \sum_{j=1}^{L} [\mathbb{I}(y_j^{(i)} \neq \hat{y}_j^{(i)})] \tag{3-3}$$

Here, n denotes the amount of samples, L refers to the number of unique labels, $y_j^{(i)}$ is the true label and $\hat{y}_j^{(i)}$ is the predicted label. The hamming loss is bounded between 0 and 1, where 0 means perfect predictions and 1 means that all predictions are wrong. This is a straightforward metric that only counts if a data point has been clustered incorrectly.

Adjusted Rand Index (ARI) The Rand Index (RI) compares the ground truth to the resulting clustering assignment. It does this by looking at how many data points are clustered correctly by checking for the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). The rand index is than calculated as follows:

$$\text{RI} = \frac{TP + TN}{TP + FP + FN + TN} \tag{3-4}$$

A RI of value of 0 means that there is no similarity between the ground truth and the clustering result, whilst a score of 1 indicates exact recovery. The downside of this method is that it does not account for random agreement of the clustering result with the ground truth.

The adjusted rand index (ARI) [51], tackles this problem and is calculated using Equation 3-5.

$$\text{ARI} = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]} \tag{3-5}$$

This ARI score has a range between -1 and 1, where 0 indicates random agreement, 1 perfect agreement and negative values indicate that the resulting clustering is worse than what would be obtained through random clustering.

3-2 IMS Dataset: Mouse pup

Spectral clustering will also be applied on one real-world IMS dataset of a 1 week old C57BL/6 control mouse pup [52]. The original data file consisted of individual spectra each containing between 10,000 and 100,000 centroid peaks that span the m/z range of 300-1,200 which consisted of 221,888 bins. The provided dataset was already pre-processed and TIC normalised by L.G. Migas PhD. This dataset consisted of 164,808 pixels with 879 bins, where the bins contained m/z values between 550-1200. Additionally, there were acquisition masks provided by L.E.M. Tideman PhD, separating the mouse pup from the background. However, upon inspection there were still multiple zero vectors present throughout in the area of the acquisition mask (Figure 3-6). The zero vectors were removed from the dataset by adding them as background pixels.

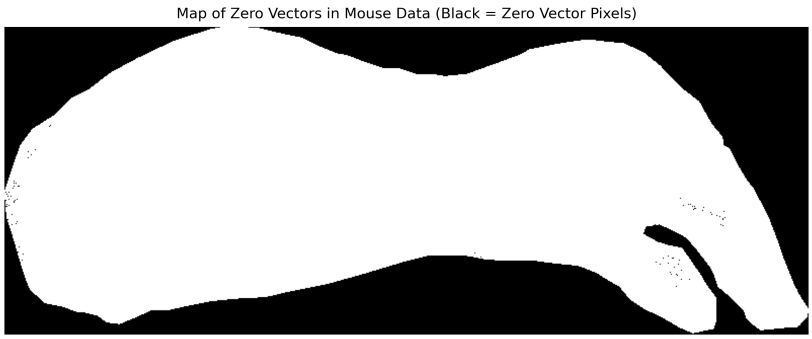


Figure 3-6: The black regions indicate the presence of zero vectors, whilst the white region show registered m/z-values at the measurement locations.

The cleaned up dataset of the mouse pup contains 163,863 data points within the acquisition mask, making the application of spectral clustering on the current setup impossible with a memory complexity of $O(n^2)$. Fortunately, an acquisition mask of the brain area was also provided, and is used for easier experimentation with hyper parameters due the significant reduction in data points ($n = 25{,}031$).

3-2-1 Mouse pup

In Figure 3-7, four different ion images of the mouse pup are presented. The complete mouse pup image contains 163,863 data points and 879 m/z-bins. The different organs within the mouse pup are highlighted at varying m/z-values, showing that the clustering should show distinction between different regions of interest. However, certain organs have high intensities at the same m/z-value, such as at $m/z = 652.328$ in the top left image, where the liver and the tongue of the mouse pup both show high intensity.

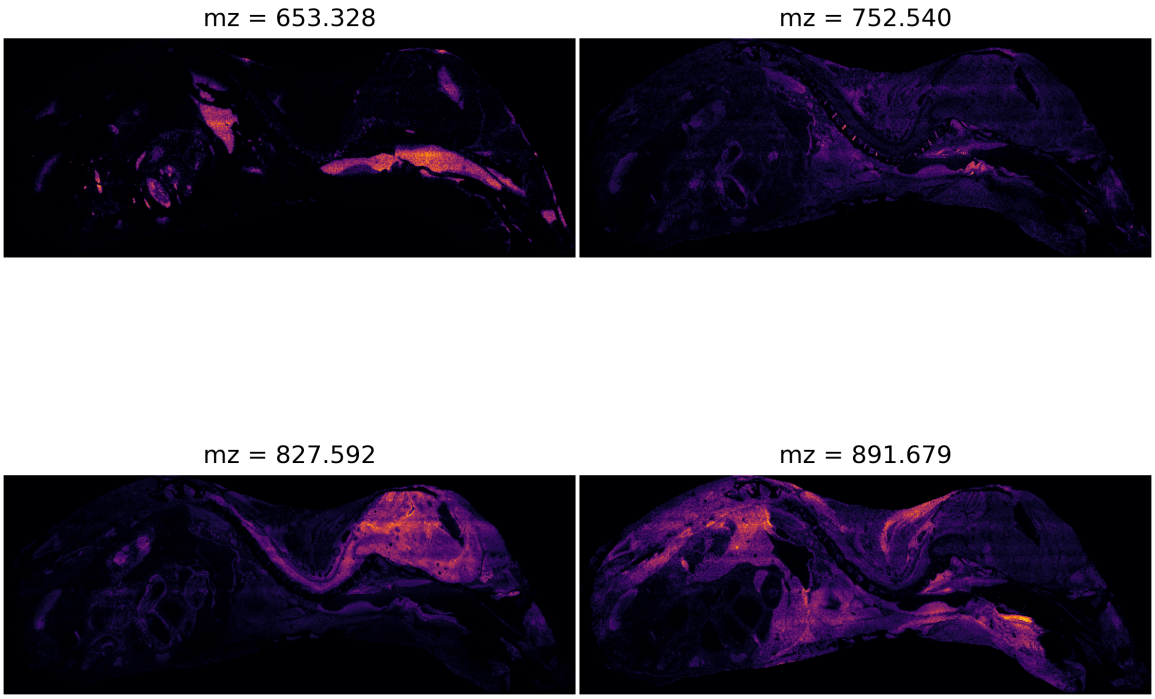


Figure 3-7: Different ion images from the mouse pup IMS dataset, a brighter color means a larger intensity at the m/z-value.

3-2-2 Mouse Brain

The dataset of the mouse brain is acquired through the brain mask as mentioned earlier. In Figure 3-8, several m/z slices of the mouse brain are shown. The reduced dataset consists of $n = 25,031$ pixels and 879 m/z bins.

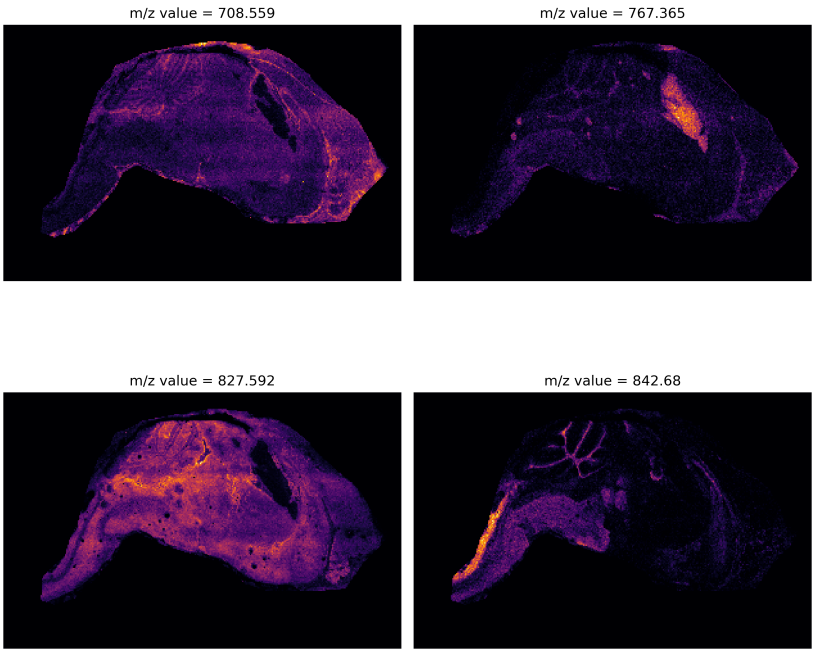


Figure 3-8: Mouse brain ion images at different m/z-values, a brighter color means a larger intensity at the m/z-value..

3-2-3 Experiment 2: Hyperparameter Selection on IMS Data.

The hyper parameters of spectral clustering are the amount of desired clusters (k) and the metrics of interest. Therefore, in this experiment a comparison is once again done between spectral clustering and (spherical) k-means, using the same similarity metrics and a varying number of clusters k. Although some conclusion could be drawn about the performance of similarity measures from the previous experiment, it is deemed appropriate to confirm this on the real IMS dataset as well.

The results will be validated both quantitatively and qualitatively. The qualitative assessment primarily focuses on ensuring that clustered regions do not exhibit irregular labeling or other inconsistencies. Again, a comparison against k-means will be provided to analyze the differences between the obtained clustering.

The results will be validated using internal validation metrics, without the presence of the ground truth as the IMS datasets are unlabeled. This is customary for clustering evaluation, where the labels are compared with the data set itself to conclude if the obtained results are satisfactory. A total of three internal validation metrics will be used.

Calinski-Harabasz Index (CHI) The Calinski-Harabasz index is defined as follows for a dataset X [53]:

$$\text{CHI} = \frac{\text{tr}(B_k)}{\text{tr}(W_k)} \cdot \frac{n_X - k}{k - 1} \tag{3-6}$$

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T \tag{3-7}$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_X)(c_q - c_X)^T \tag{3-8}$$

Where W_k represents within cluster dispersion matrix constructed as shown in Equation 3-7, B_k the between-cluster dispersion matrix constructed as shown in Equation 3-8, n_X the number of datapoints in dataset X, k the number of clusters, C_q the set of points in cluster q and c_X the center of dataset X.

If the CHI is higher, it means that there is a large distance between the clusters as B_k would contain higher values, whilst the within cluster connection is compact meaning smaller values for W_k. The advantage of this method is that it allows for quantification of clustering results without needing a ground truth, meaning that it is an internal evaluation scheme. Unfortunately, this method favors spherical convex clusters more than non-convex clusters. This can be seen by looking at the equation for W_k and B_k that utilize the Euclidean distance, which is minimal for spherical data. Consequently, methods ,such as k-means, will receive a high CHI score even if they are unable to capture non-convex clusters.

Davies-Bouldin Index (DB) The Davies-Bouldin index is defined as shown in Equation 3-9 [54].

$$\text{DB} = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij} \tag{3-9}$$

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{3-10}$$

Where, R_{ij} is a similarity measure, i and j denote different clusters, s_i is the average distance between the centroid of cluster i and the data points in cluster i and d_{ij} is the distance between the cluster centroids of i and j.

Values close to the lowest attainable score of 0 indicate a better partitioning in the DB index. This is because, the cluster centroids would be well separated and the average distance in each cluster would small.

Similarly to the CHI, the DB index does not need a ground truth. However, the current formulation of DB index introduces similar problems as the CHI index, where convex cluster shapes will obtain a low score even if they do not capture the correct clustering.

Silhouette Score (SIL) The silhouette score is another internal evaluation score. The equation for the silhouette score of a single sample is provided in Equation 3-11 [55].

$$s = \frac{b - a}{\max(a, b)} \tag{3-11}$$

Where a is the mean distance between a sample and all other points in the same class and b the mean distance between a sample and all points in the next nearest cluster. For a set of samples, the silhouette score is calculated as the average of the single sample scores. Silhouette scores can take on values between -1 and 1, where 1 indicates dense clustering and -1 wrong clustering results.

In addition, to being used as a quantitative metric, the silhouette score is also often utilized to determine the number of clusters for clustering methods [18, 56].

Unfortunately, the silhouette score also favors convex shaped clusters over non-convex ones. This is most likely due to a, which should be minimized to obtain a high score and is achieved with convex clusters.

3-2-4 Experiment 3: Memory Constrained Spectral Clustering

After conclusions are drawn for the hyperparameters from the mouse pup brain dataset, the memory constrained spectral clustering algorithm will be tested. The number of subsets r is set to 3 as this is maximum size the setup can accommodate. This means that each subset will consist of 54,621 points. The results will be constructed using the similarity measure that performed best in experiment 2 and a comparison will follow with its k-means counterpart.

The number of clusters in each subset has to remain the same, meaning that the total amount of clusters is equal to $r * k$. Therefore, k will also be varied to analyze if a constant number of clusters in each subset is a viable option to obtain interpretable results.

Validation of results The results will be validated the same way as described in subsection 3-2-3. The main difference being that the qualitative validation could include a comparison to that of the previous section, as the brain could be contained in a subset if r is chosen large enough.

Hardware Specifications As experiment 3 is designed, such that spectral clustering can run on regular workstations, the specification of the current system will be outlined. The experiments are all ran on a Lenovo Legion 5 Pro 16ACH16H, with a AMD Ryzen 7 5800H processor, 32 GB of installed RAM and a 1 TB Samsung NVMe SSD.

Chapter 4

Results

4-1 Experiment 1: Performance of Spectral Clustering in the Presence of Noise

In this section the results of the experiment 1 described in subsection 3-1-1 will be presented. This will be done two fold with qualitative and quantitative validation, the latter will be done using the adjusted rand index (ARI), hamming loss (HL) and confusion matrices as external validation metrics. The synthetic dataset was generated with the SMM model described in section 3-1, using a scaling factor $\alpha \sim U(0, 1.5)$ and a varying value for the gaussian noise ($\epsilon \sim \mathcal{N}(0, \sigma^2 I)$) with values for σ between [0, 2.0].

4-1-1 Results using Cosine Similarity and Distance

First, we will compare spherical k-means against spectral clustering using cosine metrics. The visual clustering results for a $\sigma = 0$ are portrayed in Figure 4-1. It can be seen that perfect recovery of the ground truth has been achieved by both methods, up to a permutation in labels. This is also confirmed by the validation metrics in Figure 4-2.

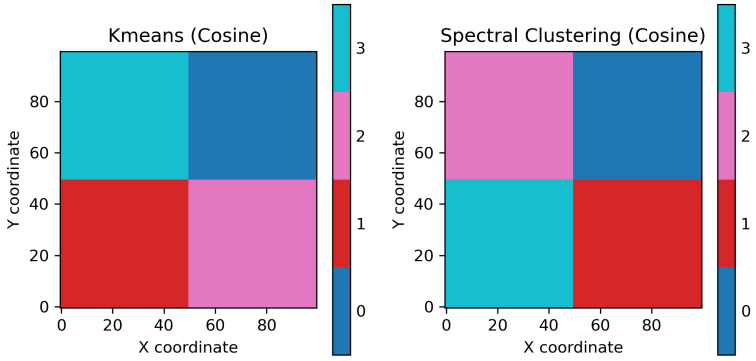


Figure 4-1: Predicted labels of the regions for $\sigma = 0$.

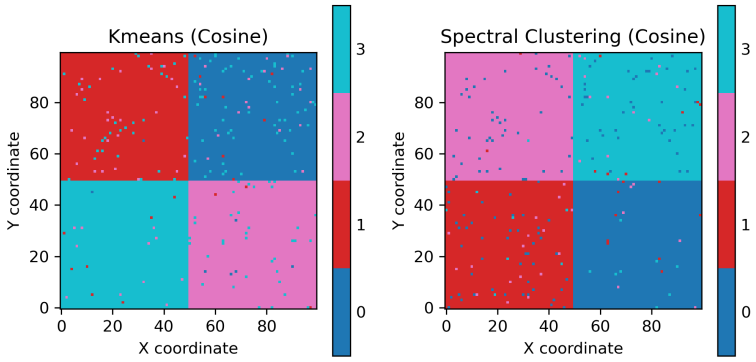


Figure 4-3: Predicted labels of the regions for $\sigma = 0.4$.

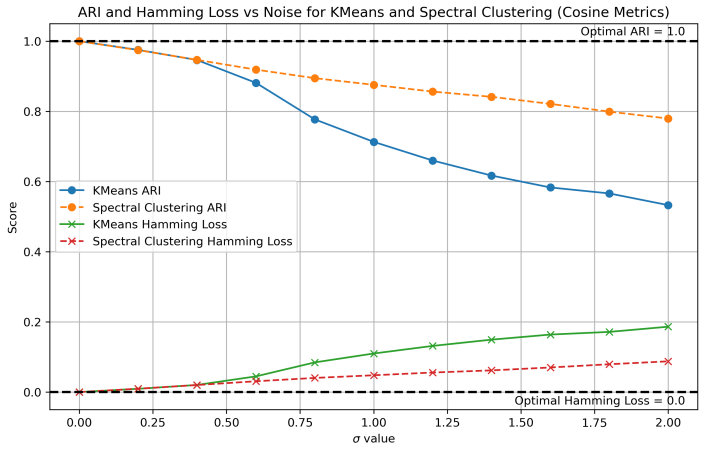


Figure 4-2: ARI and Hamming Loss for spherical k-means and spectral clustering.

However, from Figure 4-2 it is also apparent that with increasing values of σ, the hamming loss of k-means is worse than that of spectral clustering, which is also reflected in the ARI. For the current synthetic dataset, it seems that spherical k-means degrades in performance for a $\sigma > 0.4$. When looking at the predicted labels for a $\sigma = 0.4$ in Figure 4-3, the results seems to be similar in appearance. There are clear misclustered labels in each region due to the combination of signal variability and increased noise.

When this value of noise is increased, it can be noted that the performance of spectral clustering seems to degrade at a slow linear rate, whilst spherical k-means experiences a faster decrease in performance. Analyzing Figure 4-4 with a $\sigma = 2.0$, it can be noted that the predicted labels differ quite a lot now compared to Figure 4-3. Instead of misclustered labels in each region due to the noise and signal variability, it seems that spherical k-means has allocated the majority of the pixels to one of the clusters. By looking at the image, it is hard to draw a conclusion if the amount of misclustered labels by spherical k-means are equivalent to that off spectral clustering. However, the hamming loss in Figure 4-2 shows that spectral clustering has less misclustered labels. This is further confirmed when looking at the confusion matrices in Table 4-1.

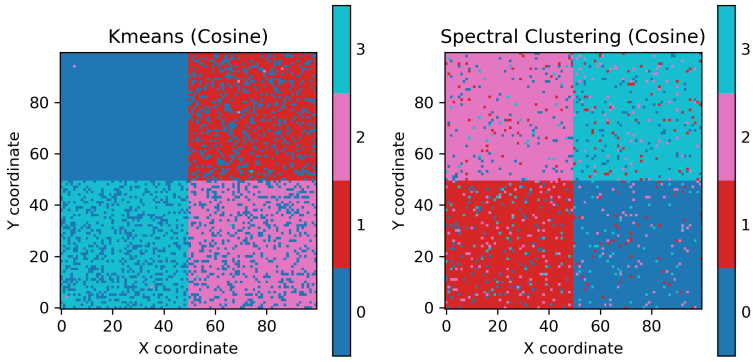


Figure 4-4: Predicted labels of the regions for $\sigma = 2.0$.

Table 4-1: Spherical k-means and spectral clustering confusion matrices for σ = [0.4, 1.0, 2.0].

k-means $\sigma = 0.4$	Predicted Label			
True Label	1	2	3	4
1	2481	11	7	1
2	35	2457	3	5
3	32	26	2441	1
4	56	18	8	2418

Spectral Clustering $\sigma = 0.4$	Predicted Label			
True Label	1	2	3	4
1	2429	47	17	7
2	9	2477	6	8
3	2	44	2453	1
4	7	40	13	2440

k-means $\sigma = 1.0$	Predicted Label			
True Label	1	2	3	4
1	2500	0	0	0
2	351	2149	0	0
3	378	0	2121	1
4	367	2	0	2131

Spectral Clustering $\sigma = 1.0$	Predicted Label			
True Label	1	2	3	4
1	2326	99	54	21
2	20	2436	22	22
3	16	88	2390	6
4	21	80	30	2369

k-means $\sigma = 2.0$	Predicted Label			
True Label	1	2	3	4
1	1822	2	676	0
2	4	1923	573	0
3	0	1	2499	0
4	1	4	602	1893

Spectral clustering $\sigma = 2.0$	Predicted Label			
True Label	1	2	3	4
1	2214	124	108	54
2	47	2047	453	49
3	52	0	2500	30
4	66	96	74	2264

4-1-2 Results using Euclidean Similarity and Distance

The visual results when using the squared euclidean metric with a $\sigma = 0$, are displayed in Figure 4-5. Both clustering algorithms were only able to reconstruct one region, but it seems as that the $\alpha \sim U(0, 1.5)$ parameter, which scales the true signal, is an issue when using the euclidean metrics. This is not surprising, as the euclidean distance is based on magnitude only. If within the same region observations are scaled, the euclidean distance between two observations can already be high.

The reason the misclustered labels are assigned to one cluster in the case of k-means can be attributed to the minimization of the distance with respect to a centroid. The centroids are iterated in each step to minimize the cost of the objective function, meaning that one centroid is drawn out by the data points that have high distances to minimize the cost of the overall objective function.

For spectral clustering, the reason can be attributed to the construction of the similarity graph. If there are many signals that have a large distance within their regions, a dense cluster will form of signals with low similarity value. When computing the eigenvectors these

points are assigned to be close to one of the graph Laplacian structures embedded in the eigenvectors. k-means than clusters these eigenvectors and clusters the data points with low similarity from the different regions to be part of the wrong region.

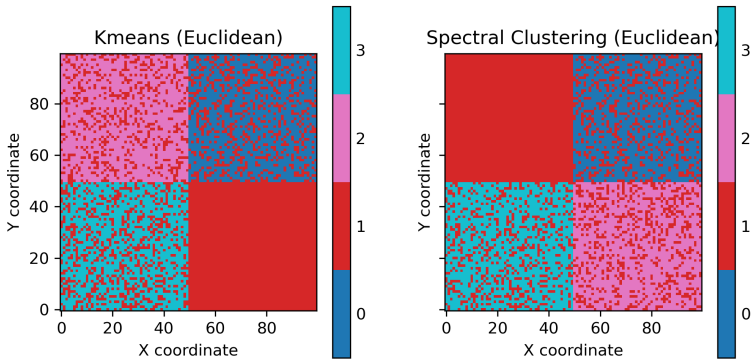


Figure 4-5: Predicted labels of the regions for $\sigma = 0.0$.

When looking at an image with a $\sigma = 2.0$ in Figure 4-6, the results seem to be equally bad visually. However, when analyzing the validation metrics in Figure 4-7 it can be noted that the result of spectral clustering started to improve with an increase in σ. This result can be linked back to what was discussed above. An increase in noise means that within a region, pixels can have a higher similarity as the noise is added independent and identically distributed (i.i.d.). This improvement is not something positive, as it is noise dependent improvement and if the noise was lower, as illustrated, the results were worse, which is counterintuitive and not desirable.

The reason why within k-means no improvement is noted, is due to the dependence on centroid whilst spectral clustering embeds the graph in eigenvectors. These results can also be analyzed within the confusion matrices shown in Table 4-2.

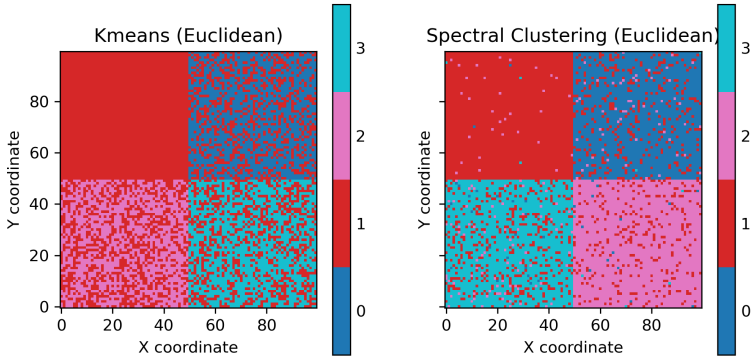


Figure 4-6: Predicted labels of the regions for $\sigma = 2.0$.

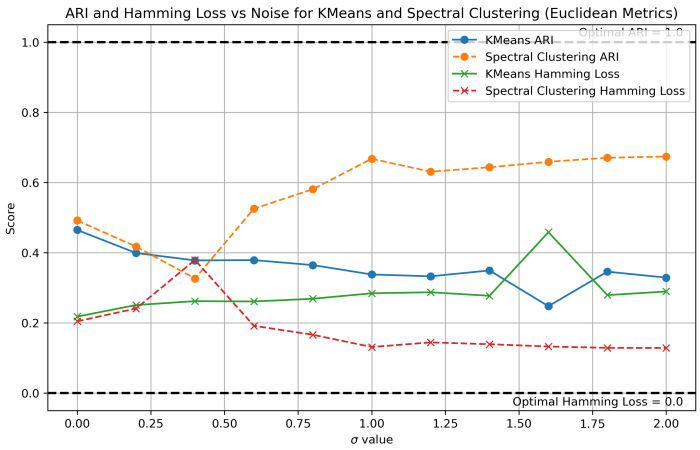


Figure 4-7: ARI and Hamming Loss for k-means and spectral clustering using euclidean metrics.

Table 4-2: k-means and spectral clustering confusion matrices for $\sigma = [0.0, 1.0, 2.0]$.

k-means $\sigma = 0.0$	Predicted Label			
True Label	1	2	3	4
1	1571	0	929	0
2	0	1696	804	0
3	0	0	2500	0
4	0	0	776	1724

Spectral Clustering $\sigma = 0.0$	Predicted Label			
True Label	1	2	3	4
1	2500	0	0	0
2	835	1665	0	0
3	791	0	1709	0
4	784	0	0	1716

k-means $\sigma = 1.0$	Predicted Label			
True Label	1	2	3	4
1	1451	291	1049	0
2	0	1594	906	0
3	0	0	2500	0
4	0	0	889	1611

Spectral Clustering $\sigma = 1.0$	Predicted Label			
True Label	1	2	3	4
1	1899	566	35	0
2	0	2497	3	0
3	0	201	2299	0
4	0	500	8	1992

k-means $\sigma = 2.0$	Predicted Label			
True Label	1	2	3	4
1	1442	0	1058	0
2	0	1565	935	0
3	0	0	2500	0
4	0	0	902	1598

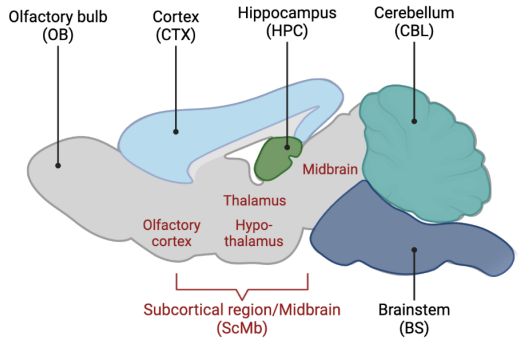
Spectral clustering $\sigma = 2.0$	Predicted Label			
True Label	1	2	3	4
1	1974	80	426	20
2	5	2194	287	14
3	3	31	2311	2
4	2	53	362	2083

4-2 Experiment 2: Hyperparameter Selection on IMS Data.

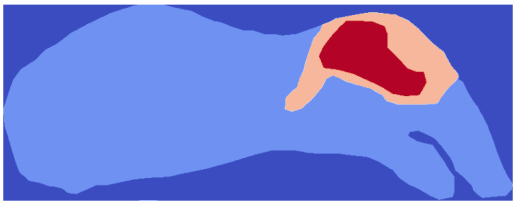
In this section the results of the second experiment outlined in subsection 3-2-3 will be presented. The number of clusters (k)for spectral clustering and (spherical) k-means were varied between [2,7]. The obtained clustering results will first be analysed qualitatively, to see if biological regions can be identified. After this a look will be taken at the quantitative results, to see if they align with the observations. Each part will only contain relevant images for the discussion, the resulting clustering assignments for both algorithms can be found in Appendix A.

4-2-1 Qualitative validation: Brain Data Set

For qualitative validation to be relevant, the different regions in the mouse brain have to be visualised first. In Figure 4-8a, a simplified view of different anatomical areas in the mouse brain are visualised to provide a guideline for what could be seen in the obtained clustering results.



(a) Anatomical areas in the mouse brain. [Image from [57]]



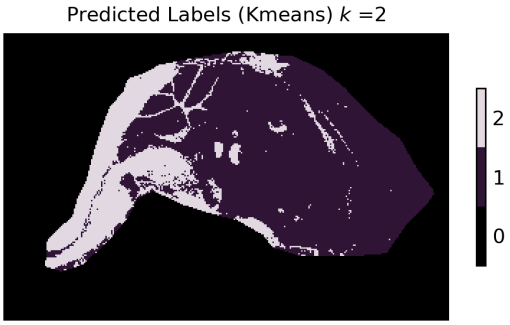
(b) Acquisition mask of the brain region, where the red pixels are definitive areas of the brain, whilst the pink pixels are border pixels that are difficult to label as it is uncertain which organ they belong to.

Figure 4-8: Anatomical areas of the mouse brain and the brain acquisition mask.

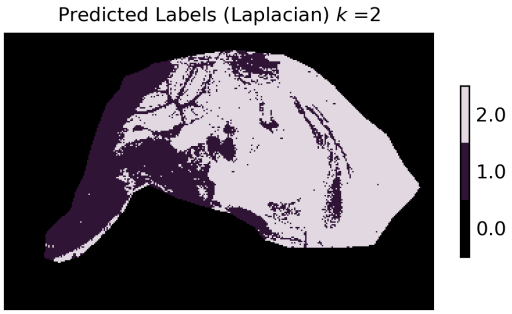
Cosine Metric Let us first discuss the results of spectral clustering using the cosine similarity and spherical k-means. In Figure 4-9, the results for $k = 2$ are presented and it can be noted that both results are quite similar. The cerebellum is captured by both clustering techniques, whilst k-means captured the brain stem in more detail. The region that should contain the hippocampus is still clustered as an uniform area, but when increasing to $k = 3$ this region is clustered as a separate area (see Figure 4-10).

For $k = 4$, the results are relatively similar to $k = 3$ where new clusters are mostly formed in the region surrounding the cortex and the area in grey displayed in Figure 4-8a (Olfactory bulb and Subcortical region/Midbrain), which also shows more definition in the area containing the hippocampus. When looking at the results for $k = 5$ in Figure 4-11, the brain stem seems to be captured better by spectral clustering but the remaining clusters seem rather speckled. In contrast, spherical k-means captures the brain stem worse but has more heterogeneous areas overall.

Finally, a look is taken at $k = 7$ in Figure 4-12b with particular interest in the results obtained by spectral clustering. The cerebellum area seems to be quite well defined, especially when comparing to a stained image of a mouse of the same sort (C576BL/6) shown in Figure 4-12a. Here the foldings and the area containing white matter seemed to be captured in more detail. This does come at the cost of fragmented clusters surrounding the cerebellum, but in none of the images was spherical k-means able to achieve this result.

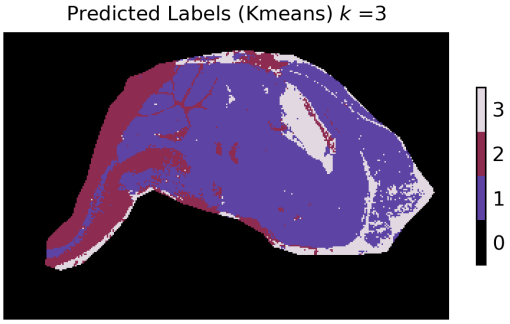


(a) Predicted labels using spherical k-means, with $k = 2$.

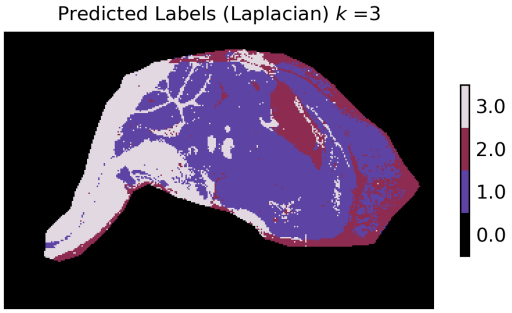


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 2$.

Figure 4-9: Predicted labels.

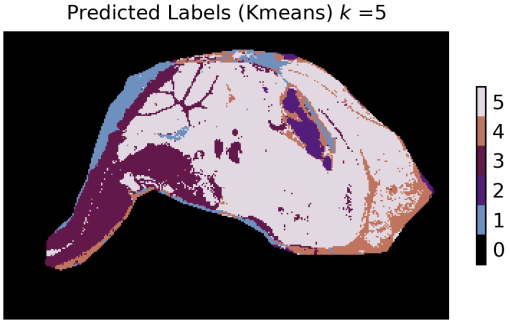


(a) Predicted labels using spherical k-means, with $k = 3$.

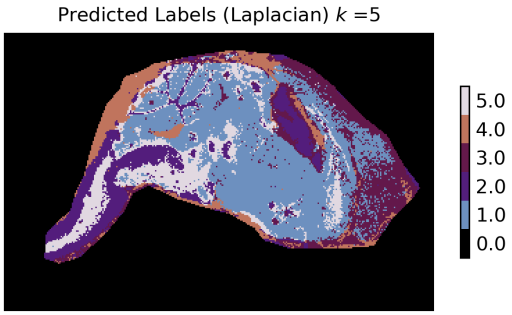


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 3$.

Figure 4-10: Predicted labels.

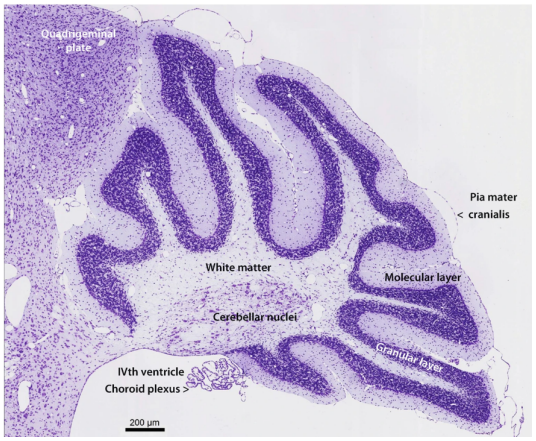


(a) Predicted labels using spherical k-means, with $k = 5$.

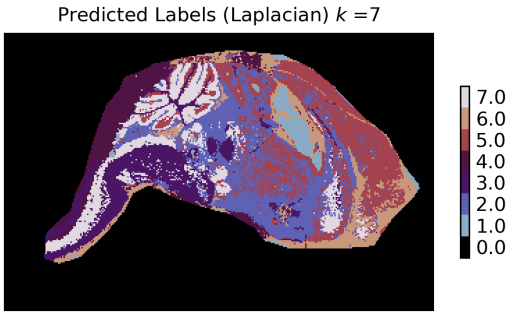


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 5$.

Figure 4-11: Predicted labels.



(a) Stained parasaggital view of the mouse brain. [Image from [58]]



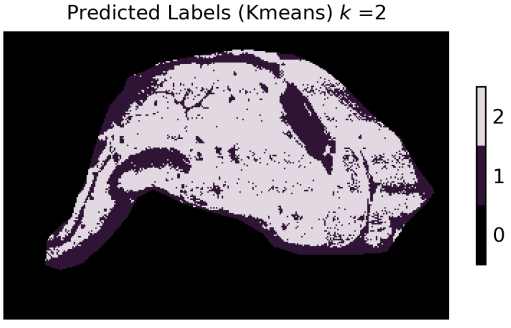
(b) Predicted labels using spectral clustering with cosine similarity, with $k = 7$.

Figure 4-12: Stained image of cerebellum and predicted labels.

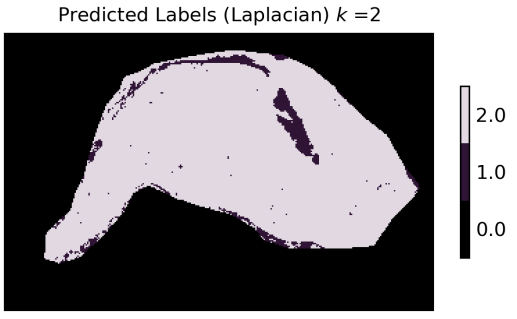
Euclidean Metric In Figure 4-13, the resulting cluster assignments are presented for spectral clustering using the euclidean similarity and k-means. These results are not similar to one another as was the case with the cosine similarity. The results obtained with k-means are quite detailed, with the brain stem and hippocampus region being clearly present. Additionally, vague outlines of the cerebellum are also visible. On the other hand, the results obtained by spectral clustering show mostly one homogeneous region, with an area of the hippocampus region outlined.

However, the results obtained by k-means quickly degraded as is visible in Figure 4-14, where horizontal lines are present through the area of the brain. The results for spectral clustering are more representative of the brain structure and have resemblance to the ones obtained by k-means for $k = 2$.

Increasing the number of clusters accentuated the horizontal lines, but the brain stem and the region containing the hippocampus remained a well clustered area for nearly all number of clusters. The outline of the cerebellum remained visible but the clusters surrounding it are all fragmented and heterogenous, see Figure 4-15.

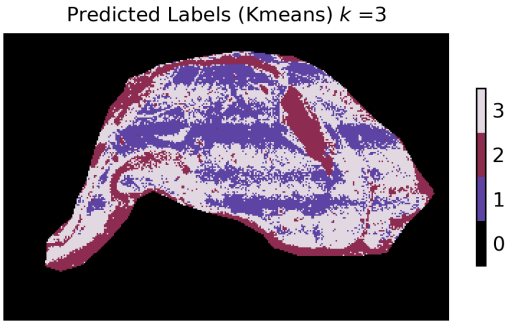


(a) Predicted labels using k-means, with $k = 2$.

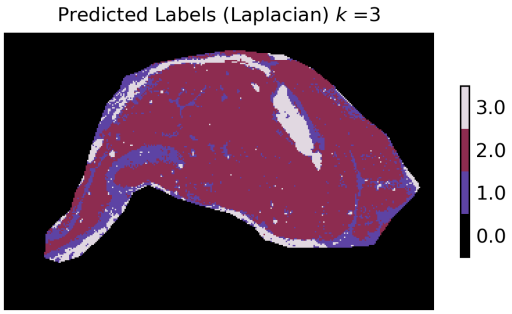


(b) Predicted labels using spectral clustering with euclidean similarity.

Figure 4-13: Predicted labels.

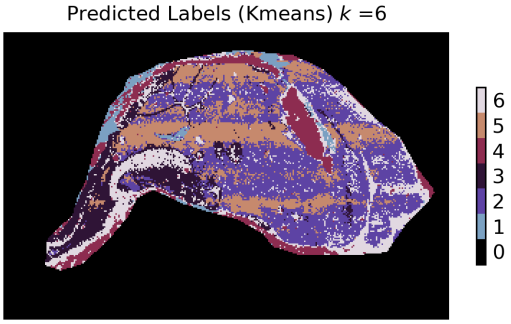


(a) Predicted labels using k-means, with $k = 3$.

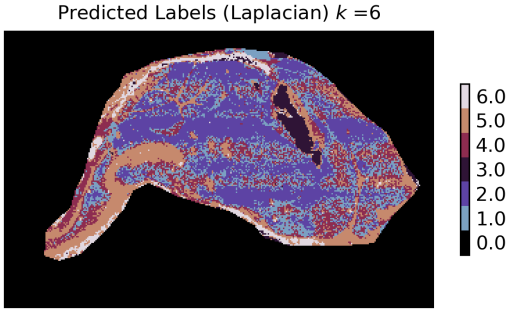


(b) Predicted labels using spectral clustering with euclidean similarity.

Figure 4-14: Predicted labels.



(a) Predicted labels using k-means, with $k = 6$.



(b) Predicted labels using spectral clustering with euclidean similarity.

Figure 4-15: Predicted labels.

4-2-2 Quantitative validation: Brain Data Set

From the qualitative validation, the initial indication is that using the cosine metric provides better visual results, resembling anatomical relevant regions. In this section, the internal validation metrics quantifying the quality of the obtained clustering results will be discussed.

We will first look at the Davies-Bouldin (DB) scores shown in Figure 4-16, a score closer to 0 indicates a better quality of obtained clusters. It can be seen that the metric favors k-means using the euclidean distance for all number of clusters, whilst spectral clustering using the cosine similarity scores the worst. However, as was observed previously for $k \geq 3$, the results obtained by k-means show fragmented clusters and heterogeneous regions. This preference for k-means has to do with how the metric is derived, as is explained in subsection 3-2-3.

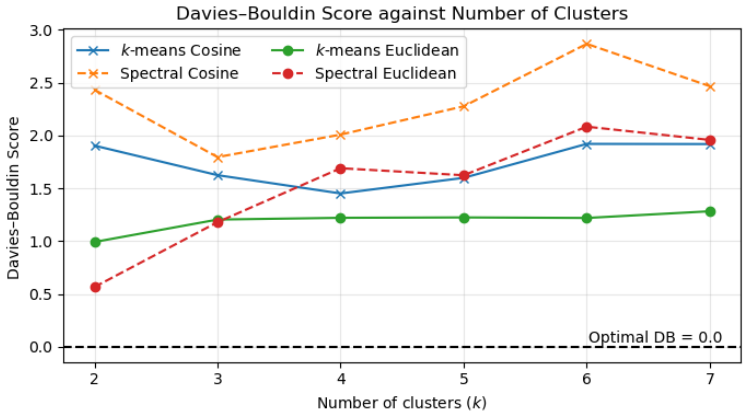


Figure 4-16: Davies-Bouldin score for different number of clusters k.

The Calinski Harabasz (CHI) score for different number of clusters is displayed in Figure 4-17, where a higher score indicates a better quality of clusters. Similarly to the DB score, the CHI index favors k-means using the euclidean distance and scores spectral clustering using the cosine distance as the worst. This time, there is also a clear difference between the alleged quality of the clusters obtained by spectral clustering using the euclidean similarity and spherical k-means. This is because the CHI score actually contains the euclidean distance in its metric to asses the resulting cluster assignment, which means that it would favor results based on euclidean distance (subsection 3-2-3).

A similar trend can be observed for the silhouette score shown in Figure 4-18, where a score closer to 1 is better, but this time more nuanced. Spectral clustering using the cosine similarity performs the worst again, but not by much. For most cluster numbers the results seem to have a similar value, but none of them show a value close to 1, with the highest value of 0.4 being achieved at $k = 2$.

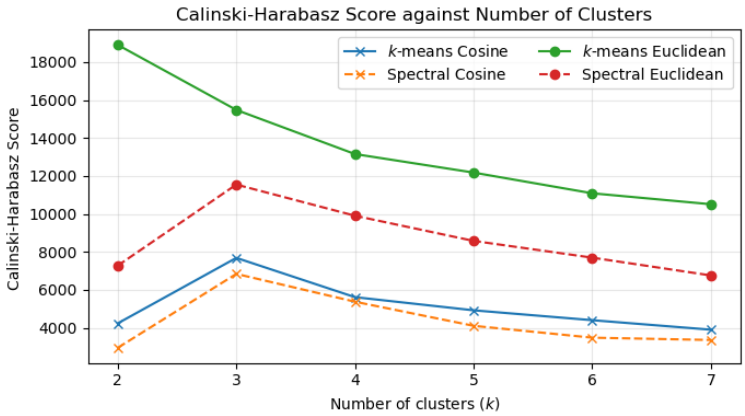


Figure 4-17: Calinski Harabasz score for different number of clusters k.

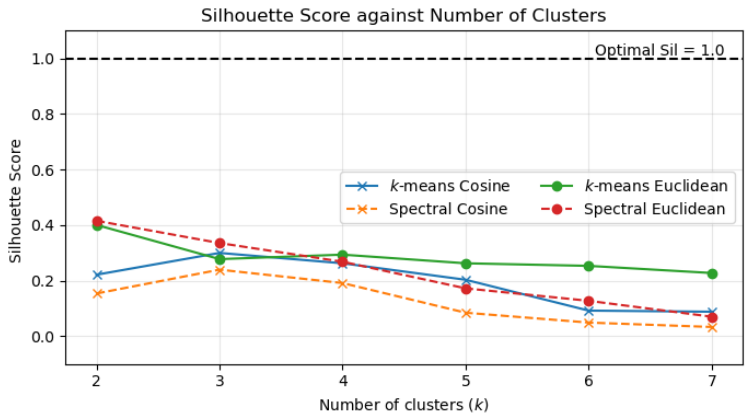


Figure 4-18: Silhouette score for different number of clusters k.

4-2-3 Discussion

The qualitative and quantitative validation are in stark contrast one another. By looking at the results visually, it is pretty apparent that for $k \geq 3$, the cosine metric provides regions that are more homogenous and anatomically relevant, whilst the euclidean metric provides mostly fragmented clusters, and contains some anatomically relevant areas.

However, when looking at the quantitative results only one would expect much better performance from the euclidean metric, and specifically k-means. The internal validation metric do have to be taken with less importance, as they are derived based on assumptions that generally favor clusters obtained by k-means. Therefore, it is deemed that the cosine metric should be used instead of the euclidean metric, as for any cluster number higher than 2, fragmented clusters start to appear rapidly.

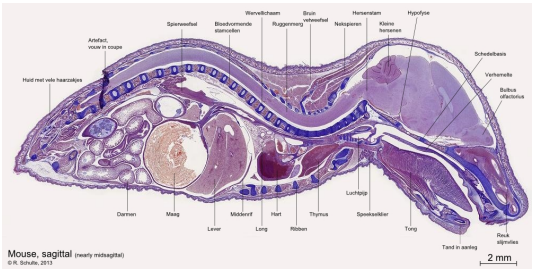
Additionally, a short note will be provided on the runtime of each algorithm. The complexity of both algorithms is highlighted in section 2-3, but the runtime was significantly longer for spectral clustering. Where k-means finished within a minute, spectral clustering took about 15-20 minutes. Given that the results are not that different, it might be preferred to use spherical k-means, if the data is of a similar form the current data.

4-3 Experiment 3: Memory Constrained Spectral Clustering

In this section, the results of the experiment presented in subsection 3-2-4 will be provided. Again a qualitative validation will be presented first, followed by a quantitative validation of the results. It should be noted that the results are only provided using the cosine metrics, as it was concluded to be the better metric for the current dataset in the discussion of experiment 2. The result for all number of clusters is provided in Appendix B.

The comparison will contain images of the different subsets ($r = 3$) and the clustering assignments for each of these subsets using spectral clustering with the cosine similarity. Additionally, the complete ims data set of the mouse is also run using spherical k-means to see if the results are similar.

Furthermore, for the complete mouse pup the liver mask, shown in Figure 4-19b, that was included in the dataset can be useful to detect if it has been included in a subset in its entirety. Additionally, a stained image of a mid sagittal cut of a mouse has been provided in Figure 4-19a, to better understand certain uniform clusters within the clustered images.



(a) Stained image of a mid sagittal cut of a mouse pup. [Image from [59]]

(b) Acquisition mask of the liver region, where the red pixels are definitive areas of the liver, whilst the pink pixels are border pixels that are difficult to label as it is uncertain which organ they belong to.

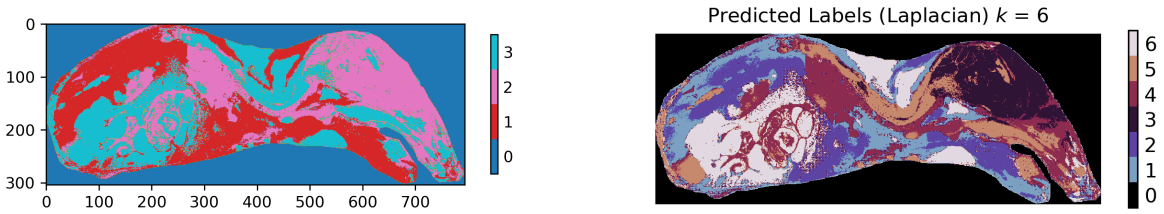
Figure 4-19: Anatomical areas of the mouse pup and the mouse pup acquisition mask.

4-3-1 Qualitative Validation

Before the images are analyzed, some information will be provided about the setting up the experiment. Each run used different random seeding point, meaning that the construction of each subset is unique. This is done this way because spectral clustering is an unsupervised learning technique and selecting the seeding location means that one has knowledge about the specimen that will be observed. Furthermore, the total number of clusters k was determined by multiplying the number of clusters per subset (rk_r), where k_r is constant for all subsets.

The results for $k = 6$, are shown in Figure 4-20, it can immediately be noticed that the liver is fragmented into different subsets (see Figure 4-20a). This results in clustering results using spectral clustering that are rather uninformative in this region. Fortunately, the brain region has been captured in its entirety, and the obtained clustering assignment is similar to the one displayed in Figure 4-10b, which shows that if the random seeding provides good subsets it could provide relevant clustering results.

Looking at the results obtained with spherical k-means, that used the complete dataset in Figure 4-21, it can be noted that the liver region has been captured here in close resemblance to the outline of liver mask. However, there is not the same level of detail in the brain region, as there is no subset division in the brain region when running spherical k-means. Nevertheless, this is at the benefit of the overall clustering results that shows homogeneous regions and clear region separation.



(a) Subset division with random seeding $r = 3$.

(b) Predicted labels using spectral clustering with cosine similarity, with $k = 6$.

Figure 4-20: Subsets and predicted labels.

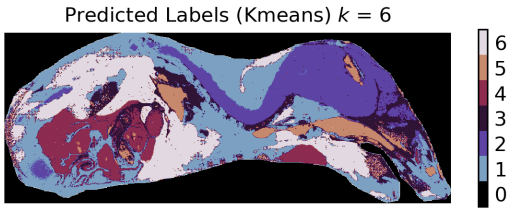
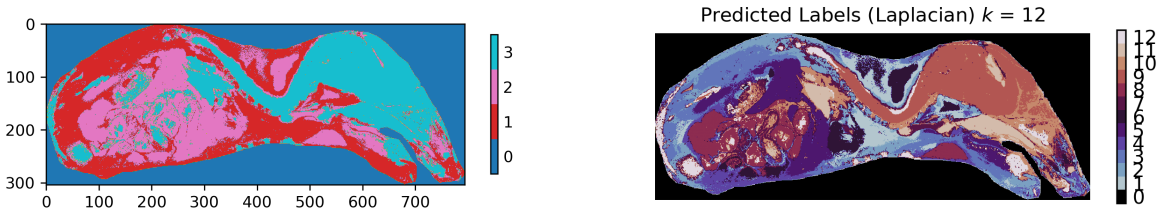


Figure 4-21: Predicted labels using spherical k-means, k=6.

We will now analyse, the results for clustering assignments that have more favorable subsets for the identification of different regions visually. In Figure 4-22, the results are presented for $k = 12$, where the subset division also specifically includes the brain region and liver region in their entirety into two separate subsets. The attained clusters, using the memory constrained version of spectral clustering, show that the liver region has been captured in its entirety, and it omits many surrounding points that were included in the subset division. Furthermore, the brain area is captured again and this time the result shows a more homogeneous region with only the region containing the hippocampus being clustered differently. Although by no means of expert opinion, when comparing the obtained clusters for $k = 12$, the lungs and the heart (indicated in Figure 4-19a with longen and hart) seem to have been captured in separate clusters as well.

Comparing it once again to the result obtained with spherical k-means in Figure 4-23, the liver region is captured similarly but it seems to be its separate cluster in its entirety, whilst in the case of spectral clustering it shares a cluster with other regions. The brain is also captured with more detail with the brain stem being visible and vague outlines of the cerebellum appearing. However, something that is captured clearer in the case of spectral clustering seems to be the spine of the mouse when different disks are quite visible. Additionally, when looking at the cluster that contains the tongue of the mouse, both labels have clustered similar regions to that cluster.



(a) Subset division with random seeding $r = 3$.

(b) Predicted labels using spectral clustering with cosine similarity, with $k = 12$.

Figure 4-22: Subsets and predicted labels.

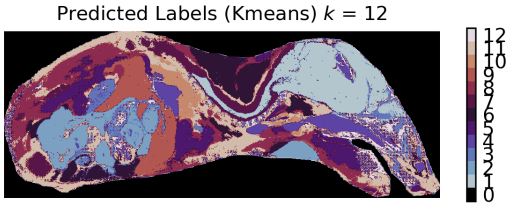


Figure 4-23: Predicted labels using spherical k-means, k=12.

For higher cluster numbers, the region containing the liver turned into two separate clusters but not as displayed in the liver mask (see Appendix B). The brain area was clustered in better detail with higher cluster numbers, however it was difficult to differentiate other areas.

4-4 Quantitative Validation

The quantitative validation will feature both spherical k-means and the memory constrained version of spectral clustering using the cosine similarity. This time the goal of the quantitative validation is not to differentiate which method performs better, but if the metrics for the different clustering assignments are somewhat in agreement.

Looking at DB score in Figure 4-24, spectral clustering seems to show more promising scores for higher cluster numbers ($k \geq 15$), whilst spherical k-means is preferred for the lower ones ($k \leq 12$). Nevertheless, the values for the DB score are rather high for all number of clusters, indicating that the clustering results are not that good. On the other hand, the CHI score indicates better quality of cluster for spherical k-means overall for all cluster number or equivalent evaluation to spectral clustering, see Figure 4-25. Finally, the silhouette score fluctuates slightly around 0 which would indicate that clusters are overlapping.

These validation metrics have to again be taken with a grain of salt, as they are not designed to evaluate the performance of spectral clustering or spherical k-means in mind. The idea behind many of these metrics is that the intracluster connection should be higher than the inter cluster connection, but due to assumptions on the cluster geometry or the formulation of the metrics, they inherently favor cluster results that align with these assumption more.

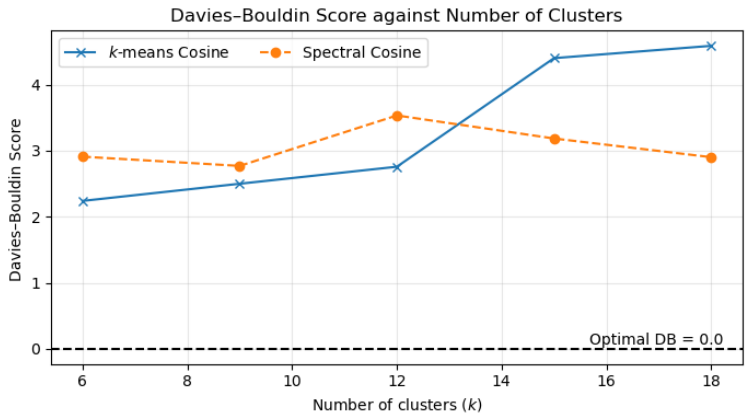


Figure 4-24: Davies-Bouldin score for different number of clusters k.

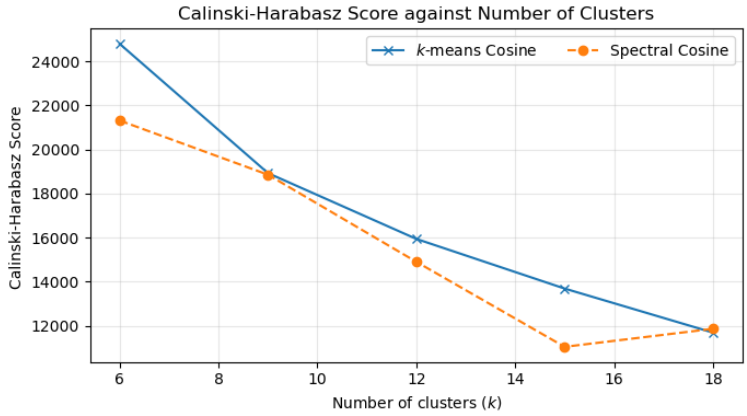


Figure 4-25: Calinski Harabasz score for different number of clusters k.

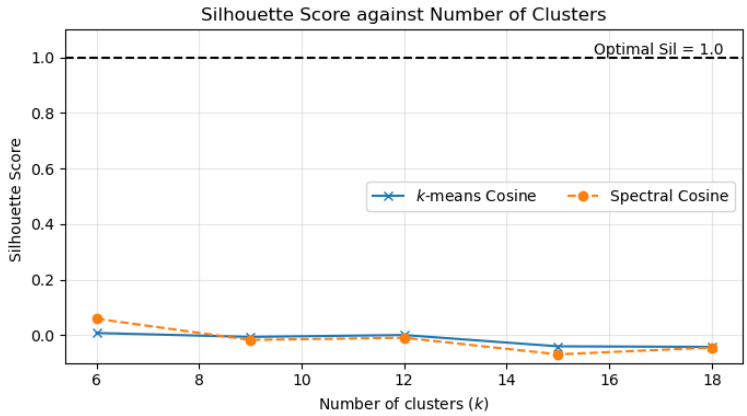


Figure 4-26: Silhouette score for different number of clusters k.

4-4-1 Discussion

The qualitative validation of the memory constrained version of spectral clustering showed that the subset division is of utmost importance to obtain relevant clustering results. The assumption that regions of interest would be captured in their entirety due to the subsets being much larger ($n/r = 54,621$)than the largest region of interest (brain with $n = 25,031$), turned out to be wishful thinking. However, in cases where the subset division did end up capturing these areas, the resulting clustering assignment looked quite good visually. Therefore, if the subset division step is updated to a more appropriate method this could end up being a viable option to apply spectral clustering on larger IMS datasets on computers that do not necessarily have the hardware specifications for it.

Nevertheless, the runtime has a much more substantial difference to that of spherical k-means this time. A singular run of the algorithm for all 3 subsets takes about 2 hours in its entirety, whilst spherical k-means finishes in about 5 to 10 minutes. The clustering results also looked highly similar to those obtained by spherical k-means, making it difficult to justify the use of spectral clustering on the current dataset.

It can very well be, that this is due the dataset being cleaned beforehand and noise not being as much of an issue, in which case the synthetic dataset confirms that the results between spherical k-means and spectral clustering should be similar. Therefore, no definitive conclusion can be made about the usefulness of spectral clustering on real world IMS datasets, as the dataset seems to not have been perturbed by noise enough to display a clear difference.

Chapter 5

Conclusion and Future Work

This work investigates the feasibility of spectral clustering in imaging mass spectrometry (IMS). The aim of the report was the provide an answer to the following research objective.

Research Objective

To develop and validate a spectral clustering framework for Imaging Mass Spectrometry (IMS) data that addresses computational scalability and resistance against noise, whilst maintaining biological segmentation.

To address each part step by step, the objective was answered with the help of three experiments. Firstly, a synthetic dataset was constructed, using a spiked mixture model (SMM), which is modified with a signal variability parameter $\alpha \sim U(0, 1.5)$ and a gaussian noise $\epsilon \sim N(0, \sigma^2 I)$ to represent IMS data. Spectral clustering was then applied using two different similarity measures, euclidean and cosine, to understand how increased noise levels and variations in ground signals within a region of interest impact the obtained clustering assignment. The obtained results were compared against k-means and spherical k-means, which showed that in every external validation metric spectral clustering using the cosine similarity performed the best for increased noise levels. However, the results of spherical k-means compared well, if not identical, against spectral clustering using the cosine similarity for lower noise levels.

This was followed by an application of the aforementioned methods on a subset of a real life IMS dataset of mouse pup which contained the brain. Here, the experimental setup was similar as the one before, where two different metric and a comparison to another clustering method was done, but the number of clusters was also a variable. Again, the clustering algorithms using the cosine metric showed more promising results qualitatively, where identified regions showed homogeneous clusters and an increased number of cluster improved the detail visible in the cluster assignment ($k \leq 4$). However, increasing the cluster number too high deteriorated the attained clusters quality. The euclidean metric, showed initially promising results at a value of $k = 2$, but with increasing cluster numbers $k \geq 3$, the clusters quickly showed fragmented clusters with horizontal splits in areas of interest.

Finally, the address the scalability of spectral clustering a memory constrained version was developed, where the complete dataset is split up into r subsets. The value r should be chosen, such that n/r is as high as possible, where n indicates the total amount of pixels. Each time the random subsets were constructed to seeding a random point and selecting the $[n/r] - 1$ closest points. This lead to different runs containing different subsets, which impacted the final clustering assignment greatly, as often regions of interest were split up into different subsets. In cases where areas were captured into subsets with entirely, the resulting cluster quality closely resembled that of spherical k-means and biological regions could clearly be identified.

An important note is that, although spectral clustering shows promising results on the synthetic dataset and real IMS datasets, the computational complexity and the memory complexity are significantly higher, to the point where it can not be recommended for use on the current real world IMS dataset. Spherical k-means works with a substantially lower amount of resources and a faster computation time, whilst providing visually similar results to spectral clustering. A proper quantitative analysis was difficult as internal validation is highly dependent on the assumptions of the designed metric, and it this scenario these did not favor the results obtained with the cosine similarity.

However, this it can not be concluded that spectral clustering has no benefit at all compared to spherical k-means, as the results of the synthetic dataset show cased that spectral clustering has superior performance in case the data is perturbed by a larger amount of noise.

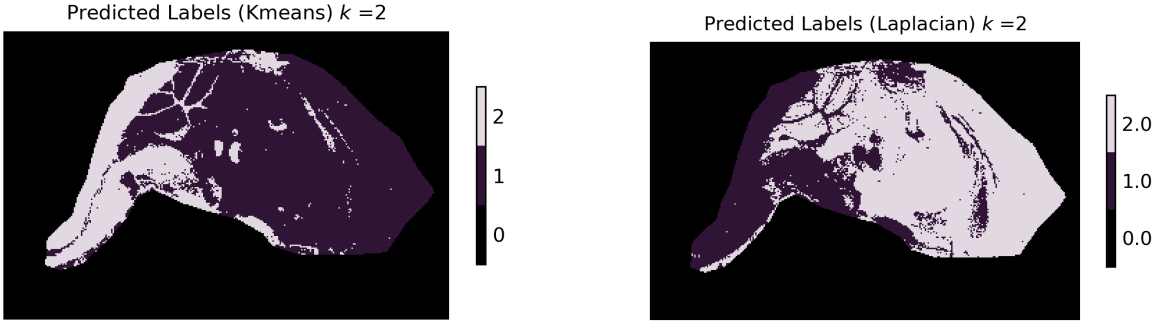
From this, several future research directions can be taken, such as:

- Analysis of spectral clustering on a more noisy real world IMS dataset.
- Develop a better seeding method than randomly selection.
- Develop an internal validation metric that can quantify the results obtained with spectral clustering.

Appendix A

Predicted Cluster Labels Brain Dataset

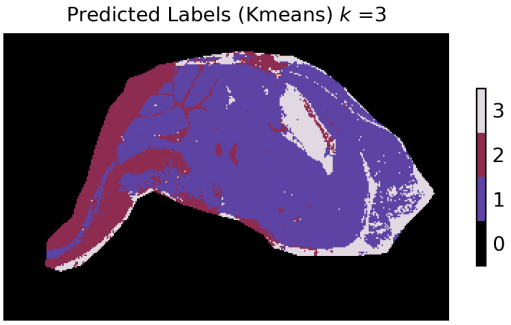
A-1 Spectral Clustering (Cosine) and Spherical k-means



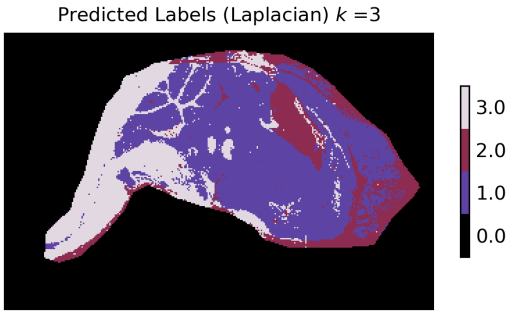
(a) Predicted labels using spherical k-means, with $k = 2$.

(b) Predicted labels using spectral clustering with cosine similarity, with $k = 2$.

Figure A-1: Predicted labels.

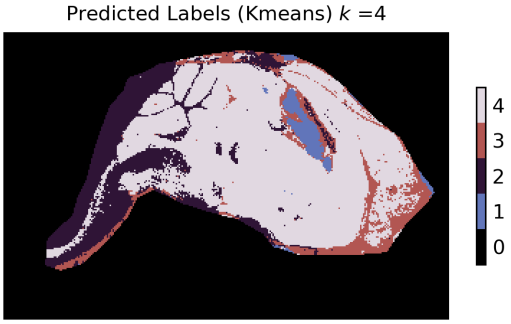


(a) Predicted labels using spherical k-means with $k = 3$.

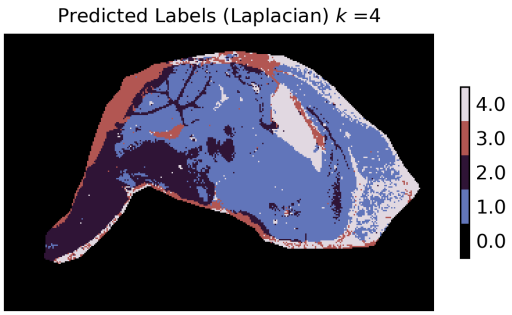


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 3$.

Figure A-2: Predicted labels.

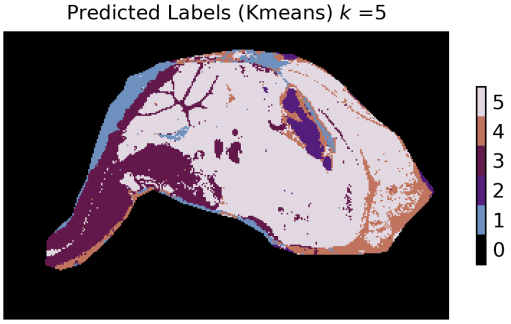


(a) Predicted labels using spherical k-means, with $k = 2$.

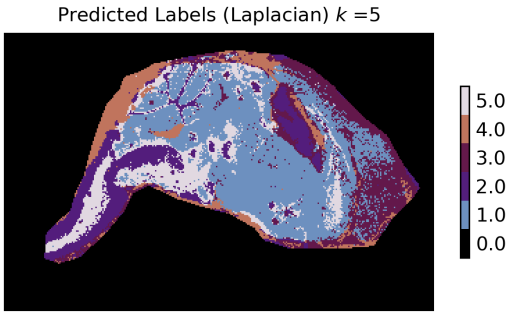


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 4$.

Figure A-3: Predicted labels.

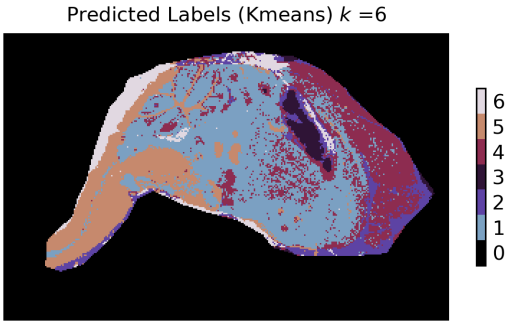


(a) Predicted labels spherical k-means, with $k = 5$.

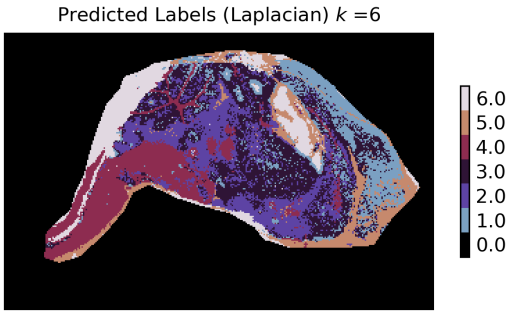


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 5$

Figure A-4: Predicted labels.

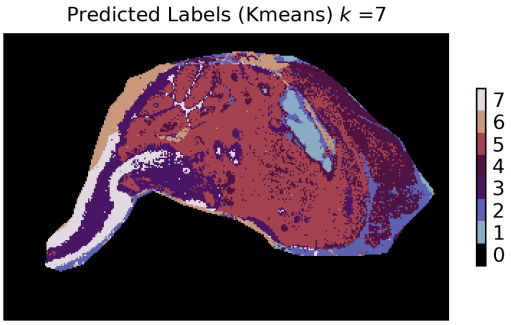


(a) Predicted labels spherical k-means, with $k = 6$.

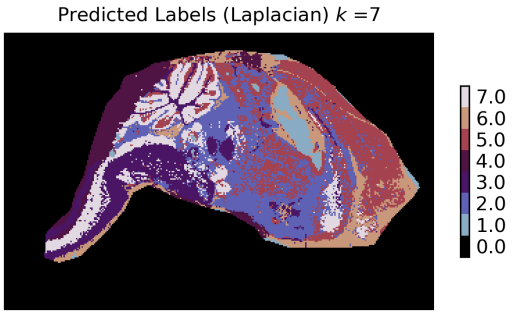


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 6$

Figure A-5: Predicted labels.



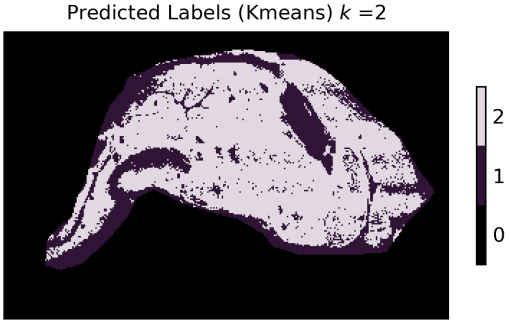
(a) Predicted labels spherical k-means, with $k = 7$.



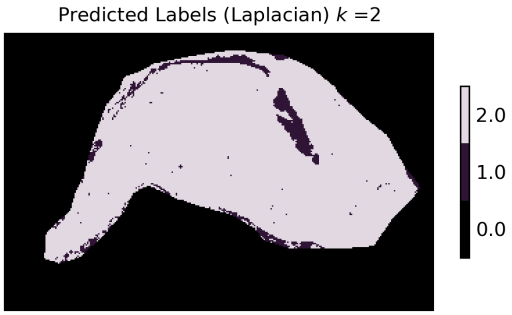
(b) Predicted labels using spectral clustering with cosine similarity, with $k = 7$

Figure A-6: Predicted labels.

A-2 Spectral Clustering (Euclidean) and k-means

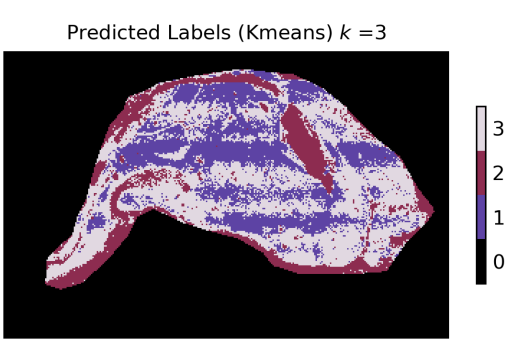


(a) Predicted labels using k-means, with $k = 2$.

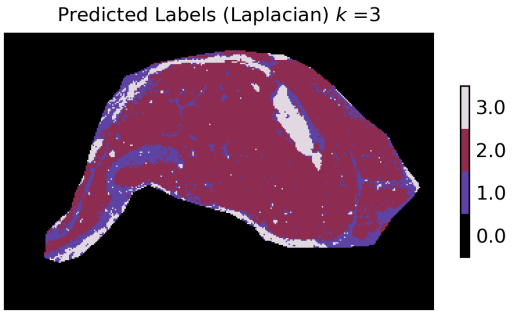


(b) Predicted labels using spectral clustering with euclidean similarity.

Figure A-7: Predicted labels.

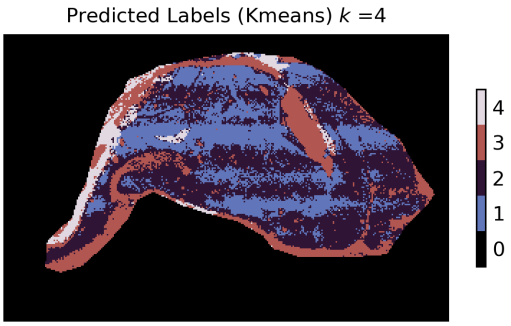


(a) Predicted labels using k-means, with $k = 3$.

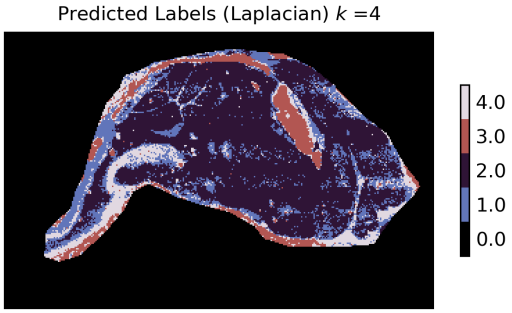


(b) Predicted labels using spectral clustering with euclidean similarity, with $k = 3$.

Figure A-8: Predicted labels.

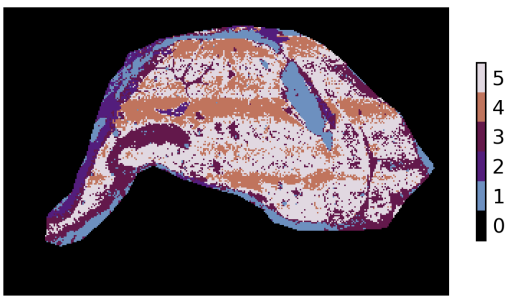


(a) Predicted labels using k-means, with $k = 4$.

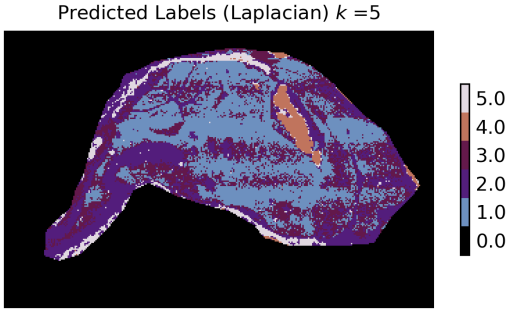


(b) Predicted labels using spectral clustering with euclidean similarity, with $k = 4$.

Figure A-9: Predicted labels.

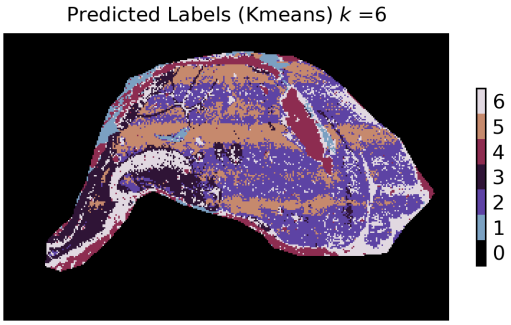


(a) Predicted labels using k-means, with $k = 5$.

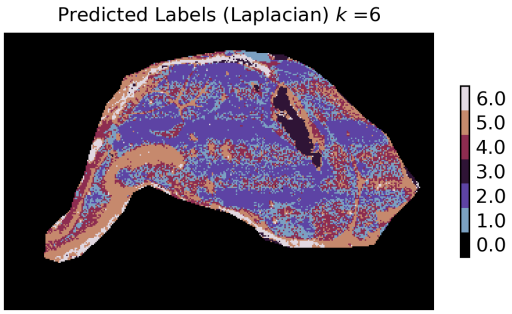


(b) Predicted labels using spectral clustering with euclidean similarity, with $k = 5$.

Figure A-10: Predicted labels.

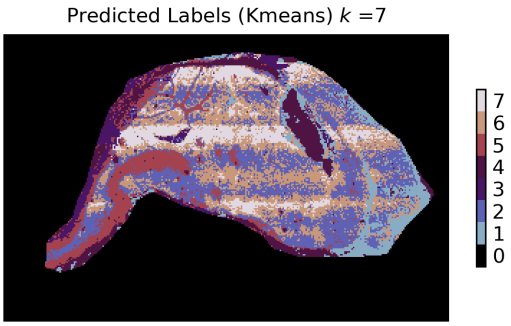


(a) Predicted labels using k-means, with $k = 6$.

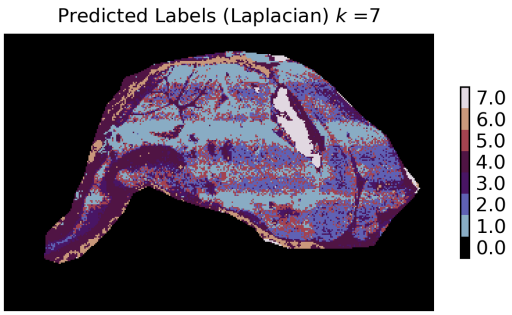


(b) Predicted labels using spectral clustering with euclidean similarity, with $k = 6$.

Figure A-11: Predicted labels.



(a) Predicted labels using k-means, with $k = 7$.



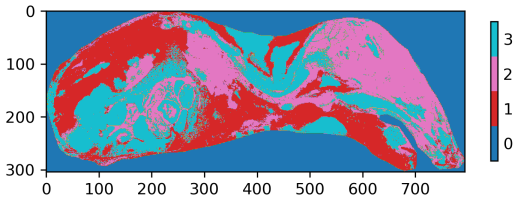
(b) Predicted labels using spectral clustering with euclidean similarity, with $k = 7$.

Figure A-12: Predicted labels.

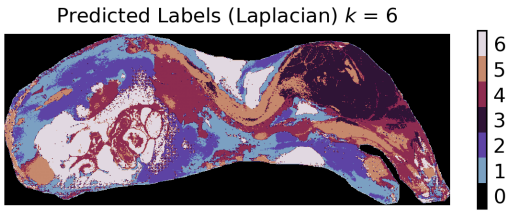
Appendix B

Predicted Cluster Labels Mouse Pup Dataset

B-1 Memory Efficient Spectral Clustering

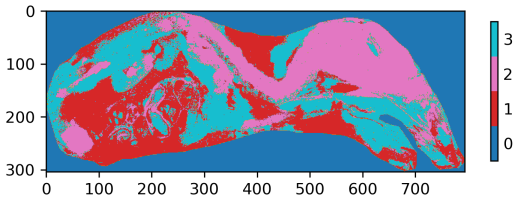


(a) Subset division with random seeding $r = 3$.

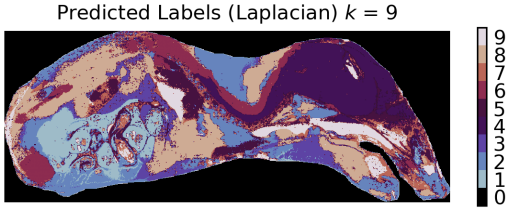


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 6$.

Figure B-1: Subsets and predicted labels.

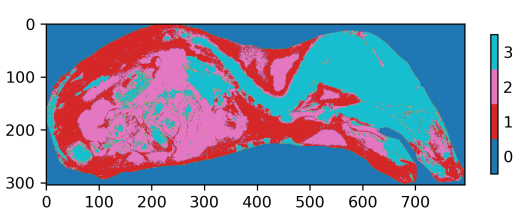


(a) Subset division with random seeding $r = 3$.

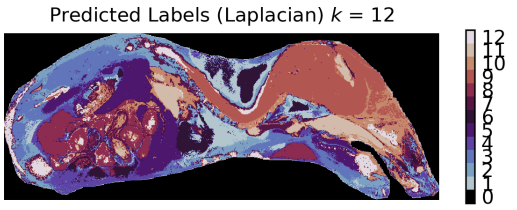


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 9$.

Figure B-2: Predicted labels.

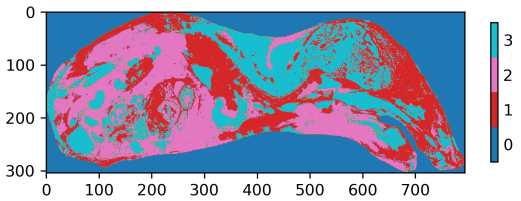


(a) Subset division with random seeding $r = 3$.

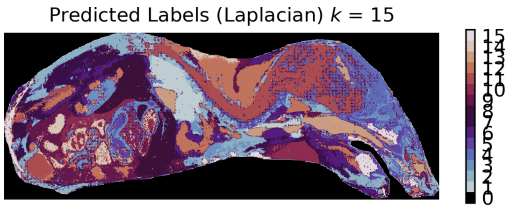


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 12$.

Figure B-3: Predicted labels.

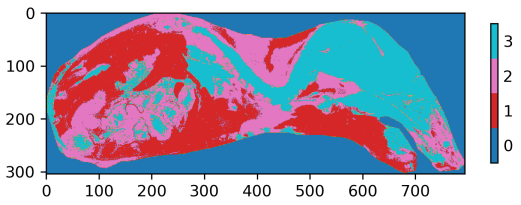


(a) Subset division with random seeding $r = 3$.

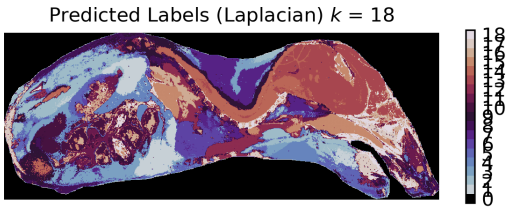


(b) Predicted labels using spectral clustering with cosine similarity, with $k = 15$.

Figure B-4: Predicted labels.



(a) Subset division with random seeding $r = 3$.



(b) Predicted labels using spectral clustering with cosine similarity, with $k = 18$.

Figure B-5: Predicted labels.

B-2 Spherical k-means

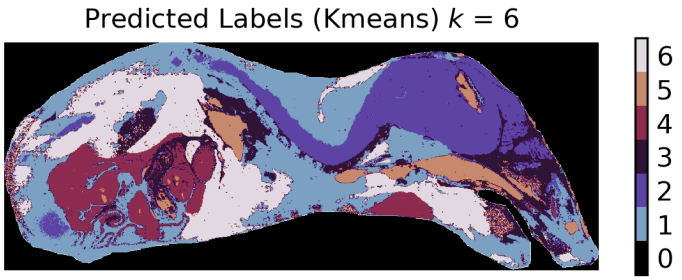


Figure B-6: Predicted labels using spherical k-means, k=6.

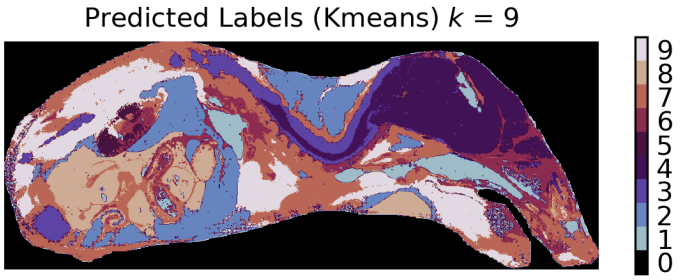


Figure B-7: Predicted labels using spherical k-means, k=9.

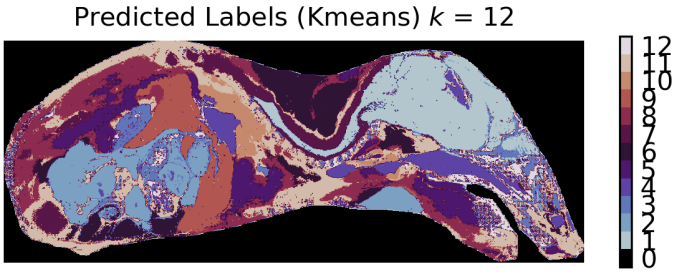


Figure B-8: Predicted labels using spherical k-means, k=12.

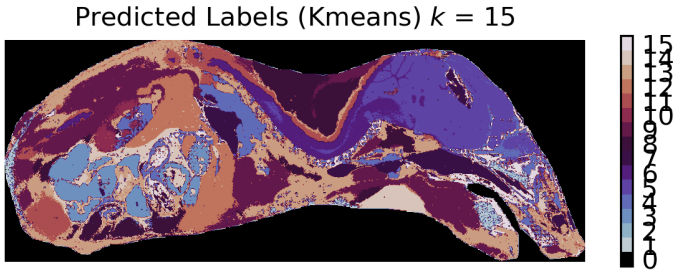


Figure B-9: Predicted labels using spherical k-means, k=15.

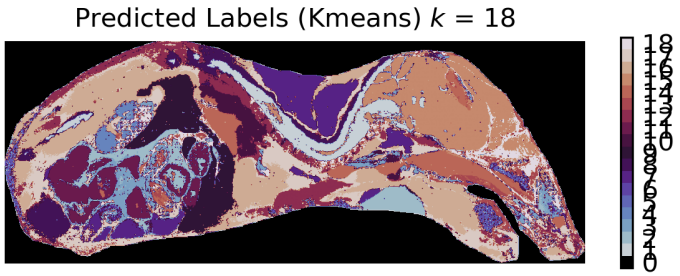


Figure B-10: Predicted labels using spherical k-means, k=18.

Bibliography

[1] Yolanda Picó. Chapter 2 - advanced mass spectrometry. In Yolanda Picó, editor, *Advanced Mass Spectrometry for Food Safety and Quality*, volume 68 of *Comprehensive Analytical Chemistry*, pages 77–129. Elsevier, 2015.

[2] Liam A. McDonnell and Ron M. A. Heeren. Imaging mass spectrometry. *Mass Spectrometry Reviews*, 26(4):606–643, 2007.

[3] Kamila Chughtai and Ron MA Heeren. Mass spectrometric imaging for biomedical tissue analysis. *Chemical reviews*, 110(5):3237–3277, 2010.

[4] Mary E King, Monica Lin, Meredith Spradlin, and Livia S Eberlin. Advances and emerging medical applications of direct mass spectrometry technologies for tissue analysis. *Annual Review of Analytical Chemistry*, 16(1):1–25, 2023.

[5] Jiangjiang Liu and Zheng Ouyang. Mass spectrometry imaging for biomedical applications. *Analytical and bioanalytical chemistry*, 405(17):5645–5653, 2013.

[6] Nicolas Verbeeck, Richard M. Caprioli, and Raf Van de Plas. Unsupervised machine learning for exploratory data analysis in imaging mass spectrometry. *Mass Spectrometry Reviews*, 39(3):245–291, 2020.

[7] Chihiro Murayama, Yoshishige Kimura, and Mitsutoshi Setou. Imaging mass spectrometry: principle and application. *Biophysical Reviews*, 1(3):131–139, 2009.

[8] Nicholas P. Lockyer, Satoka Aoyagi, John S. Fletcher, Ian S. Gilmore, Paul A. W. van der Heide, Katie L. Moore, Bonnie J. Tyler, and Li-Tang Weng. Secondary ion mass spectrometry. *Nature Reviews Methods Primers*, 4:32, 2024.

[9] Emmanuelle Claude, Emrys A. Jones, and Steven D. Pringle. Desi mass spectrometry imaging (msi). In *Methods in Molecular Biology*, volume 1618, pages 65–75. 2017.

[10] Nobuhiro Zaima, Takahiro Hayasaka, Naoko Goto-Inoue, and Mitsutoshi Setou. Matrix-assisted laser desorption/ionization imaging mass spectrometry. *International Journal of Molecular Sciences*, 11(12):5040–5055, 2010.

[11] Raf Van de Plas. Computational aspects of imaging mass spectrometry. Slide Presentation, 2018. Copyright ©2018 Vanderbilt University.

[12] Salvatore Cappadona, Fredrik Levander, Maria Jansson, Peter James, Sergio Cerutti, and Linda Pattini. Wavelet-based method for noise characterization and rejection in high-performance liquid chromatography coupled to mass spectrometry. *Analytical Chemistry*, 80(13):4960–4968, 2008. PMID: 18510348.

[13] Dante Mantini, Francesca Petrucci, D. Pieragostino, P. Del Boccio, M. Di Nicola, C. Di Ilio, Giorgio Federici, Paolo Sacchetta, Silvia Comani, and Andrea Urbani. Limpic: a computational method for the separation of protein maldi-tof-ms signals from noise. *BMC Bioinformatics*, 8(101), 2007.

[14] Deukwoo Kwon, Marina Vannucci, Joon Jin Song, Jaesik Jeong, and Ruth M Pfeiffer. A novel wavelet-based thresholding method for the pre-processing of mass spectrometry data that accounts for heterogeneous noise. *Proteomics*, 8(15):3019–3029, 2008.

[15] Jeremy L. Norris, Dale S. Cornett, James A. Mobley, Malin Andersson, Erin H. Seeley, Pierre Chaurand, and Richard M. Caprioli. Processing maldi mass spectra to improve mass spectral direct tissue analysis. *International Journal of Mass Spectrometry*, 260(2):212–221, 2007. Imaging Mass Spectrometry Special Issue.

[16] Sören-Oliver Deininger, Dale S Cornett, Rainer Paape, Michael Becker, Charles Pineau, Sandra Rauser, Axel Walch, and Eryk Wolski. Normalization in maldi-tof imaging datasets of proteins: practical considerations. *Analytical and bioanalytical chemistry*, 401(1):167–181, 2011.

[17] Sarah A. Schwartz, Robert J. Weil, Mahlon D. Johnson, Steven A. Toms, and Richard M. Caprioli. Protein profiling in brain tumors using mass spectrometry: Feasibility of a new technique for the analysis of protein expression. *Clinical Cancer Research*, 10(3):981–987, 02 2004.

[18] Laurine Lagache, Yanis Zirem, Émilie Le Rhun, Isabelle Fournier, and Michel Salzet. Predicting protein pathways associated to tumor heterogeneity by correlating spatial lipidomics and proteomics: The dry proteomic concept. *Molecular & Cellular Proteomics*, 24(1), 2025.

[19] 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. *Analytical Chemistry*, 85(23):11189–11195, 2013. PMID: 24180335.

[20] Andrew R Konicek, Jonathan Lefman, and Christopher Szakal. Automated correlation and classification of secondary ion mass spectrometry images using ak-means cluster method. *Analyst*, 137(15):3479–3487, 2012.

[21] M. Prasad, G. Postma, P. Franceschi, et al. Evaluation and comparison of unsupervised methods for the extraction of spatial patterns from mass spectrometry imaging data (msi). *Scientific Reports*, 12:15687, 2022.

[22] Sören-Oliver Deininger, Matthias P. Ebert, Arne Fütterer, Marc Gerhard, and Christoph Röcken. Maldi imaging combined with hierarchical clustering as a new tool for the interpretation of complex human cancers. *Journal of Proteome Research*, 7(12):5230–5236, 2008. PMID: 19367705.

[23] Gregor McCombie, Dieter Staab, Markus Stoeckli, and Richard Knochenmuss. Spatial and spectral correlations in maldi mass spectrometry images by clustering and multivariate analysis. *Analytical Chemistry*, 77(19):6118–6124, 2005. PMID: 16194068.

[24] Mohamed El Ayed, David Bonnel, Rémi Longuespée, Céline Castellier, Julien Franck, Daniele Vergara, Annie Desmons, Aurélie Tasiemski, Abderraouf Kenani, Denis Vinatier, Robert Day, Isabelle Fournier, and Michel Salzet. Maldi imaging mass spectrometry in ovarian cancer for tracking, identifying, and validating biomarkers. *Medical Science Monitor: International Medical Journal of Experimental and Clinical Research*, 16(8), August 2010.

[25] Wanqiu Zhang, Marc Claesen, Thomas Moerman, M. Reid Groseclose, Etienne Waelkens, Bart De Moor, and Nico Verbeeck. Spatially aware clustering of ion images in mass spectrometry imaging data using deep learning. *Analytical and Bioanalytical Chemistry*, 413(10):2803–2819, 2021.

[26] Dan Guo, Melanie Christine Föll, Kylie Ariel Bemis, and Olga Vitek. A noise-robust deep clustering of biomolecular ions improves interpretability of mass spectrometric images. *Bioinformatics*, 39(2):btad067, 2023.

[27] Michael Steinbach, Levent Ertöz, and Vipin Kumar. *The Challenges of Clustering High Dimensional Data*, pages 273–309. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[28] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006.

[29] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3(1), March 2009.

[30] Richard Bellman, Robert E Kalaba, et al. *Dynamic programming and modern control theory*, volume 81. Citeseer, 1965.

[31] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[32] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems*, pages 873–879, 2001.

[33] Anima Anandkumar, Rong Ge, Daniel Hsu, and Sham M. Kakade. A tensor approach to learning mixed membership community models. *Journal of Machine Learning Research*, 15(1):2239–2312, 2014.

[34] Francis R. Bach and Michael I. Jordan. Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, 7:1963–2001, 2006.

[35] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274. ACM, 2001.

[36] Matthias Löffler, Anderson Y. Zhang, and Harrison H. Zhou. Optimality of spectral clustering in the gaussian mixture model, 2020.

[37] Paul-Louis Delacour, Sander Wahls, Jeffrey M. Spraggins, Lukasz Migas, and Raf Van de Plas. Signal recovery using a spiked mixture model, 2025.

[38] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[39] Afonso S. Bandeira. 18.s096: Spectral clustering and cheeger's inequality. http://math.mit.edu/~bandeira, 2015. Lecture Notes for Topics in Mathematics of Data Science, MIT, October 6, 2015.

[40] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multi-way spectral partitioning and higher-order cheeger inequalities. *arXiv preprint arXiv:1111.1055*, 2011.

[41] Shiping Liu. Multi-way dual cheeger constants and spectral bounds of graphs. *Advances in Mathematics*, 268:306–338, 2014.

[42] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, pages 849–856, Vancouver, Canada, 2002. Neural Information Processing Systems Foundation.

[43] Eric Bunch. Spectral clustering, 2018.

[44] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multi-way spectral partitioning and higher-order cheeger inequalities. *STOC '12*, pages 1117–1130, 2012.

[45] Alex Dexter, Alan M. Race, Iain B. Styles, and Josephine Bunch. Testing for multivariate normality in mass spectrometry imaging data: A robust statistical approach for clustering evaluation and the generation of synthetic mass spectrometry imaging data sets. *Analytical Chemistry*, 88(22):10893–10899, 2016. PMID: 27641083.

[46] Sayantan Pal, Maiga Chang, and Maria Fernandez Iriarte. Summary generation using natural language processing techniques and cosine similarity. In *International Conference on Intelligent Systems Design and Applications*, pages 508–517. Springer, 2021.

[47] Yasunao Takano, Yusuke Iijima, Kou Kobayashi, Hiroshi Sakuta, Hiroki Sakaji, Masaki Kohana, and Akio Kobayashi. Improving document similarity calculation using cosine-similarity graphs. In *International Conference on Advanced Information Networking and Applications*, pages 512–522. Springer, 2019.

[48] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[49] Rameshwar Pratap, Anup Deshmukh, Pratheeksha Nair, and Tarun Dutt. A faster sampling algorithm for spherical k-means. In *Proceedings of The 10th Asian Conference on Machine Learning*, volume 95 of *Proceedings of Machine Learning Research*, pages 343–358. PMLR, 2018.

[50] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2):143–175, jan 2001.

[51] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[52] Jeffrey M. Spraggins, Katerina V. Djambazova, Emilio S. Rivera, Lukasz G. Migas, Elizabeth K. Neumann, Arne Fuetterer, Juergen Suetering, Niels Goedecke, Alice Ly, Raf Van de Plas, and Richard M. Caprioli. High-performance molecular imaging with maldi trapped ion-mobility time-of-flight (timstof) mass spectrometry. *Analytical Chemistry*, 91(22):14552–14560, 2019.

[53] T. Caliński and J Harabasz and. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[54] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[55] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[56] The scikit-learn developers. Selecting the number of clusters with silhouette analysis on KMeans clustering, 2025. Accessed: 2025-04-04.

[57] Natasha M. Kafai, Hana Janova, Matthew D. Cain, Yashar Alippe, Sofia Muraro, Alan Sariol, Michelle Elam-Noll, Robyn S. Klein, and Michael S. Diamond. Entry receptor ldlrad3 is required for venezuelan equine encephalitis virus peripheral infection and neurotropism leading to pathogenesis in mice. *Cell Reports*, 42(8):112946, 2023.

[58] H. Schr"oder, N. Moser, and S. Huggenberger. The mouse cerebellum. In *Neuroanatomy of the Mouse*, pages 153–170. Springer, Cham, 2020.

[59] Ronald Schulte. Mid-sagittal coupe of a mouse.