Silesian University of Technology
Faculty of Automatic Control, Electronics
and Computer Science

Politechnika
Śląska

# Clustering techniques of high-throughput big -omics data

PhD Thesis

Author: Grzegorz Mrukwa
Supervisor: prof. dr hab. inż. Joanna Polańska
Co-supervisor: dr inż. Michał Marczyk

Gliwice, July 2022

# Contents

# Chapter 1

# Introduction

Analysis of big -omics data provides unparalleled insights allowing us to understand the reasons for local and organism-wide phenomena on a chemical compound or gene level. Such insights allow hypothesizing about internal tumor heterogeneity, tumor clones, or causes of its resistance to therapies. However, the data is highly complex as it consists of tens of thousands of features, and different physical phenomena appear during the acquisition. Moreover, the techniques of big -omics data acquisition improve resolution over time, hence the amount of data growing vastly.

The scientific community acknowledges the issue and tested numerous approaches, but the gap between machine learning experts providing the tool and biologists using the tool remains unacceptable. The scalability of these tools is seriously limited, and often the explainability of the results provided by the known approaches is poor. At the same time, it is hard to numerically assess the quality of unsupervised methods.

The unsupervised analysis problem has two main components: feature space adaptation and observation clustering. Most of the research focuses either on one or another, while it is critical to match a clustering algorithm to the specifics of the feature space. An example is the widely used k-means clustering, which requires appropriate tuning to extract relevant detail. Additionally, in the dimensionality of tens of thousands of features, a curse of dimensionality occurs, which renders crucial nuances indistinguishable from the irrelevant data noise.

The most powerful feature engineering approaches aim to visualize the data set structure, which is not necessarily optimal for reusing the obtained feature space for clustering purposes. At the same time, it is hard to explain the contribution of specific features in the final representation of the data set.

## 1.1 Aim and Theses of This Work

This work aims to provide a consistent and scalable framework for knowledge discovery in different kinds of big -omics data, with a specific focus on Mass Spectrometry Imaging.

In order to create such a framework, this thesis is based on existing research, combines, calibrates, and automates classical methods. Combination of multiple existing methods allowed to achieve a more comprehensive result than the original approaches. Calibration enabled sensitivity of the methods for biological nuance recognition in a heavily multidimensional feature space. Automation eliminated the need for a manual hyperparameter search and evaluated numerous scenarios to select the optimal one.

Thanks to the accomplished work, an analytical methodology has been created for unsupervised investigation of the big -omics data, which takes into account the problem of feature engineering and clustering simultaneously. The established methodology is flexible enough for a drop-in replacement of components, which makes it incredibly easy to use with the newest computational techniques.

The framework is benchmarked against state-of-the-art methods on two different Mass Spectrometry Imaging data sets, covering a highly detailed 2D whole-tissue sample and high-throughput 3D data. Based on the obtained results, we have drawn a set of conclusions, which helped formulate the following theses as the most important results of this dissertation:

1. A stepwise methodology applied to big -omics data clustering can provide results comparable to the existing state-of-the-art one-step methods.

2. Scalability is an essential aspect for analysis of big -omics data, as the volume of the dataset grows in size both in the number of dimensions and the number of observations. A combination of highly-scalable yet simple methods provides superior scalability for the number of observations without a significant loss of the discovered level of details.

3. Deep neural networks are an efficient and scalable tool for unsupervised data analysis in big -omics. Nuances missed by the network trained on the entire dataset also can be captured in a stepwise setup, similar to one proposed for the classical feature engineering method. It is possible to construct a flexible stepwise big -omics analytical methodology, easy to update with the newest methods for increased fidelity.

## 1.2 Novel Aspect

The joint analysis of feature relevance and observations simultaneously is nothing new. In the past, algorithms like High Dimensional Data Clustering [10] were used but are inadequate for the amount of data today. The field of Natural Language Processing is experiencing a renaissance after implementing representation learning together with classification (and other tasks) in the form of transformer neural networks [86], and biomedical research could benefit from similar approaches.

Surprisingly, most of the known approaches to unsupervised knowledge discovery in big -omics data still cover either feature engineering or observation grouping. This work is supposed to address the niche of joint feature engineering and clustering.

The hierarchical clustering scheme inspires the approach proposed in this work but flexibly adapts the number of clusters at each level of the hierarchy of clusters. The feature space adaptation is embedded into the process, selecting locally relevant features at each node of the clusters' hierarchy.

## 1.3 Thesis Structure

The thesis is structured in the following way:

- In Chapter 2 we explain the origin of the data used in this work for benchmark purposes. It explains how biological information is acquired in a digital form. Describes the consequences of the selected acquisition procedure on the data quality and a preprocessing pipeline required to mitigate these issues.

- In Chapter 3, we describe the current state of the art in big -omics data clustering. The approaches are compared on a high level to provide the reader with their principle of operation, strengths, the areas of their application, and known limitations.

- In Chapter 4, we propose a robust and scalable method for big -omics data clustering, called Divisive Intelligent K-Means (DiviK). Since it is based on the K-Means algorithm, which is very sensitive to configuration, we explain how it is calibrated to work well with the datasets used in this study. We evaluate DiviK on the datasets introduced in Chapter 2 and discuss the results.

- In Chapter 5 a next step in the development of the DiviK framework is demonstrated: Divisive clustering with Variational Autoencoders. We combine the DiviK framework and the newest advances in deep learning to overcome the limitations of both methods. At the end of the chapter, we include the results of the numerical evaluation of the method.

- In Chapter 6, we summarize this work and indicate in which situations Divisive Intelligent K-Means framework could be useful. The summary is backed with a list of examples where DiviK was already successfully applied.

## 1.4 Software Implementation

For the purpose of reusability, the software implementation is conceptually separated into two components.

First is a publicly available software library for Python programming language. The library is distributed pre-compiled via Python Package Index under name `divik` (`https://pypi.org/project/divik/`) and in source code form via GitHub (`https://github.com/gmrukwa/divik`). The API of the package follows the `scikit-learn` guidelines [27].

The second component consists of multiple private repositories specific for conducting an initial exploration of a given data set and its further analysis using Divisive Intelligent K-Means. These repositories contain complete configuration and package versions locked to keep all the computations fully reproducible, thus will be shared online on demand and are attached physically to the USB drive.

# Chapter 2

# Material

## 2.1 Mass Spectrometry Imaging

Mass Spectrometry Imaging (MSI) is an emerging untargeted -omics method for analyzing molecular profiles that generates big hyperspectral imaging data. It provides an unparalleled insight into the metabolomics of the tissue sample [83, 58, 8, 9]. The high volume of the MSI data already motivated the development of dedicated computational methods, which address either metabolome analysis [41, 3, 128], hyperspectral image analysis [135, 139], or both [10, 12].

According to [9], the development of Mass Spectrometry Imaging was at an over-exponential pace, similarly to the AI domain. However, since 2015, the growth of AI has even accelerated, and a risk materializes that spatial metabolomics will miss out on the occurring AI revolution.

At the same time, the development of spatial metabolomics is powered by the urgent and increasing needs in biology and medicine to characterize the role of metabolism in health and diseases. MSI promises answers to a range of questions regarding tumor molecular profiling [132, 94, 71], immune system cells function [26], microbiota contributions to inflammation [120], metabolic dysregulations during the infection [49], and many more [51, 48, 77, 84, 111].

From the technological perspective, Mass Spectrometry Imaging is an excellent example of biological big data, with the following characteristics:

- *Volume*: the spatial resolution of a scan varies between $5-100\mu m$, which leads to 10,000 - 4,000,000 spectra potentially acquired from a $1cm^2$ tissue sample with the low-resolution ToF spectrometers, but becomes even more severe for a 3D data set [130].

- *Velocity*: the last five years brought more than 6,000 MSI datasets uploaded to the METABASE database [90, 1].

- *Variety*: a low-resolution dataset may consist of 200,000 mass channels (features) describing a single spectrum (observation), representing proteins, peptides, or metabolites. Even more channels are captured for high-resolution scans.

- *Veracity*: information in MSI data is heavily duplicated, but the duplicated patterns are often overlapping and hard to separate. Capturing the most relevant nuances is complex, with dominating patterns amplified during the data collection process [95]. Moreover, the feature importance may vary across functional tissue regions.

The process of acquiring the metabolomic information in a digital form is depicted in Figure 2.1. It explains the example of the Matrix-Assisted Laser Desorption/Ionization (MALDI) Time-of-Flight (ToF) method. The tissue sample is sliced into sections. Sections are coated with trypsin to split proteins into smaller particles like peptides and lipids, and the matrix which causes the particles to ionize. Then, the spectrometer applies a raster, and multiple laser shots in each pixel of the raster release ions from the tissue surface. Note that the laser beam destroys the tissue sample during the imaging process. The released ions are directed into a vacuum tube and accelerate in the electric field. Basic physics laws allow us to estimate the mass-to-charge ratio for each ion based on the time of its flight through the vacuum tube of known size in a fixed electric field. This way, for each pixel in the raster, we obtain a *mass spectrum* – a signal representing how many ions of a specific mass-to-charge ratio ($m/z$) were captured. An often simplification is to assume that all ions were loaded with a unit charge.

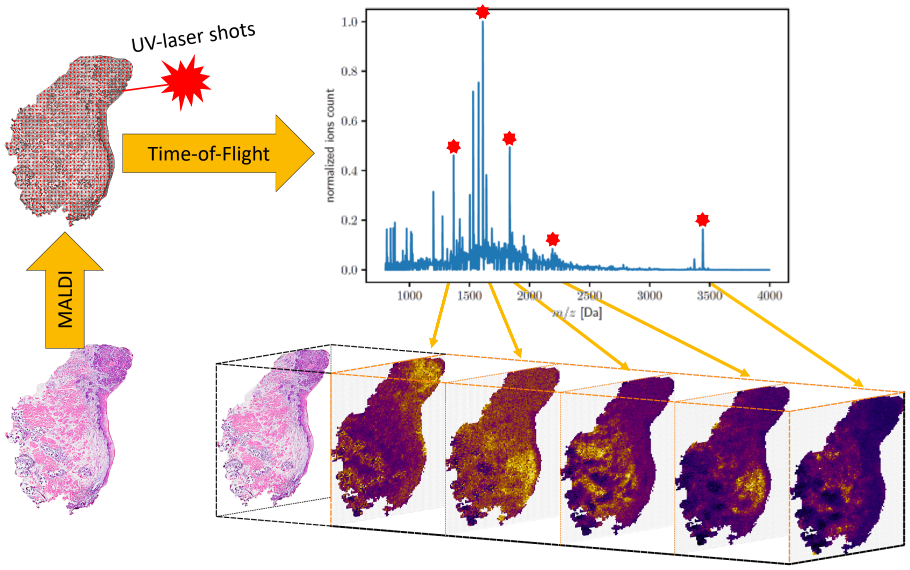


Figure 2.1: Mass Spectrometry Imaging data acquisition with MALDI-ToF. The tissue sample is coated with a matrix and subject to laser desorption. Released ions are directed to a vacuum tube and accelerated in the electric field. Ions are captured at the output, and a mass spectrum is formed. After normalization of the spectra to the common $m/z$ axis, the spatial distribution of ions with selected $m/z$ can be investigated.

The acquired mass spectra are annotated with spatial coordinates of their pixel in the raster. This spatial annotation leads to a dual interpretation of the dataset: for each mass-to-charge value, one can create a distribution map of the captured particle and visualize it across the entire tissue sample. Of course, even for a low mass resolution of the aperture, ions reaching the detector cannot be expected to perfectly align in time when they originate from different points on the scanned tissue sample. Therefore an $m/z$ axis may vary between spectra, although similar in range. Varying $m/z$ axis and many more phenomena characteristic of such a data acquisition method require careful handling during a dedicated dataset preprocessing pipeline.

Sample preprocessing pipeline may consist of [15, 79, 99]:

- *Resampling*: Each spectrum has its own $m/z$ axis with slight variations, yet analysis of the entire dataset at once requires a common $m/z$ axis. Spectra get resampled so that the $m/z$ axis is exactly the same.

- *Spectrum Smoothing*: Captured spectra are noisy, as visualized in Figure 2.2. Savitzky-Golay filter is one of the most popular methods for smoothing the spectra to reduce the high-frequency noise [99].

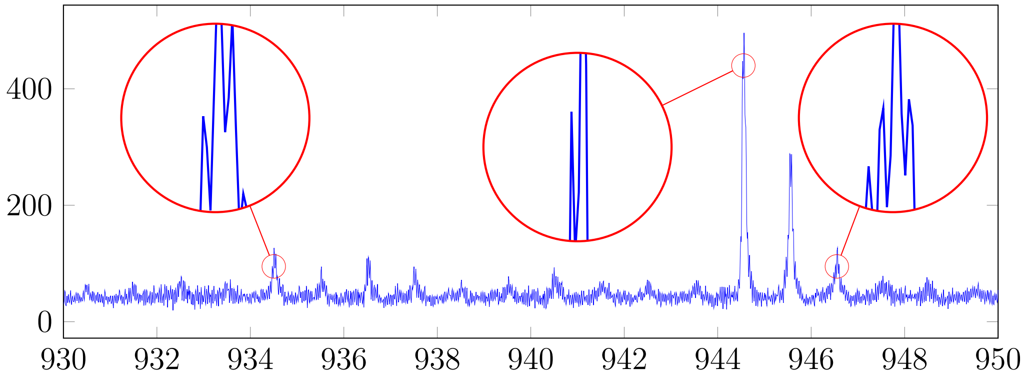


Figure 2.2: Sample noisy MSI spectrum.

- *Baseline Removal*: The matrix is a critical part of the Matrix-Assisted Laser Desorption/Ionization experiment as it assists molecules that otherwise cannot be easily ionized. However, matrix molecules can also be ionized without actually being attached to any molecule of interest. The second potential reason for baseline to occur is that the ion source gets contaminated, especially towards the end of a single MALDI ToF study [38]. Both potential causes lead to false-positive counts captured by the detector and negatively impact the signal-to-noise ratio. Figure 2.3 presents the original spectrum and the same spectrum after a simple baseline removal procedure.

- *Outlier Detection (optional)*: Spectra with extremely low and extremely high Total Ion Count (TIC) are removed from further analysis. The spectra with extremely high TIC would strongly influence the later stages of the processing, while the spectra with extremely low TIC would have their noise amplified beyond any possible interpretation. Figure 2.4 presents such sample outlier spectra.

- *Peak Alignment*: Peaks in the captured spectra are often desynchronized on the $m/z$ axis, but they still represent the same chemical compound.

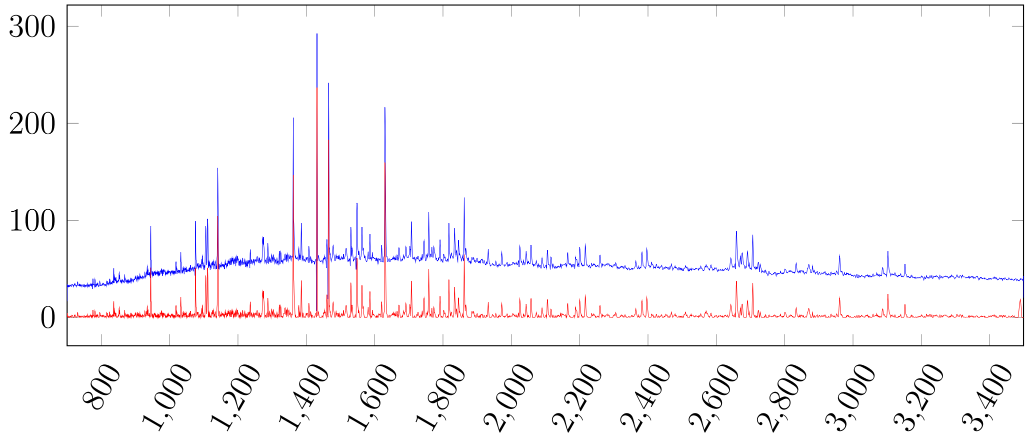


Figure 2.3: Spectra baseline removal effect. The original spectrum in blue, spectrum after baseline removal in red.

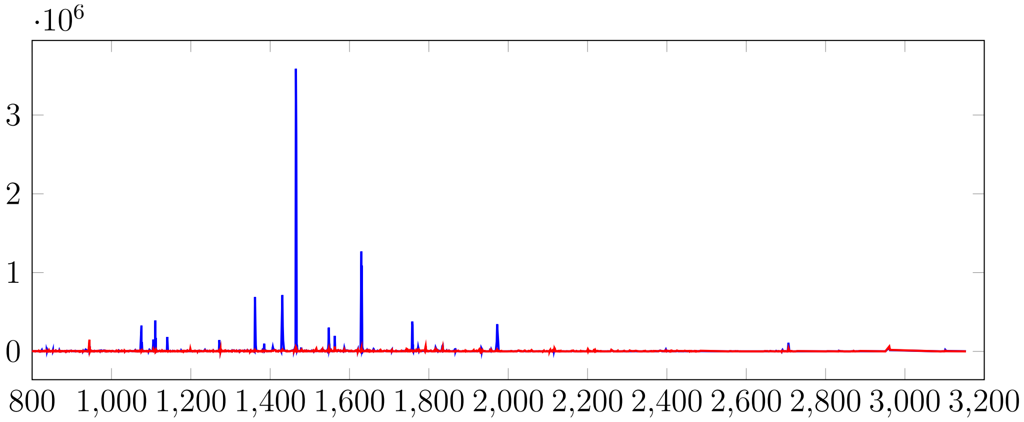


Figure 2.4: Outlier mass spectra from the same experiment with very low (red) and very high (blue) TIC.

The dependency is non-linear, and the most basic translations and scaling do not solve the issue. Figure 2.5 depicts such desynchronization. Peak alignment based on Fast Fourier Transform is a commonly used method for peak alignment, as it allows for non-linear relative peak displacements [134].

- *Spectrum Normalization*: The reasons for spectrum normalization are the same as for baseline removal. Baseline removal can correct the signal-to-noise ratio; however, it does not directly influence the scale of

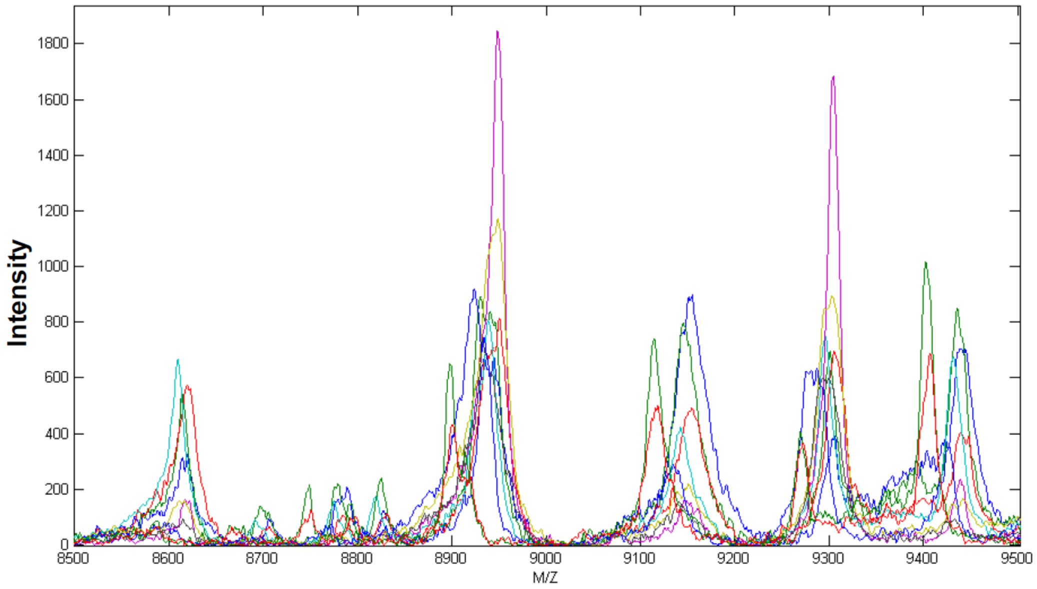


Figure 2.5: Shifts in the same mass peak captured on different spectra are often non-linear and decrease the effective mass resolution of the captured data.

detected peaks so that they are comparable across the dataset [99]. For this, we need spectrum normalization. Although outlying spectra are removed already, the spectra of varying intensity require projection onto a common intensity axis for an informative analysis. The projection is often realized by normalizing the total ion count across the entire dataset.

- *Peak Picking*: Captured spectra contain massively duplicated information. Multiple peaks may represent a single chemical compound in the MSI spectrum due to varying isotopes present in the tissue sample, slight differences in cuts occurring with the help of trypsin, and imperfections in the data acquisition hardware. Moreover, a single peak in the MSI spectrum is already characterized by numerous mass channels, which collectively contain information about the peak shape. There are many ways for picking and/or learning the peaks, starting from simplest gradient methods [138], through signal compression methods like wavelets [15] or Gaussian Mixture Model (GMM) [95], ending on the newest advances of deep learning [2].

Following the mentioned dataset preprocessing steps should reduce the negative impact of data acquisition process limitations on the final result of the numerical analyses. However, some researchers [2] claim to overcome the issue without the classical approach explained above.

To verify the hypothesis stated in this work, we use two datasets: Oral Squamous Cell Carcinoma 2D MSI dataset and Mouse Kidney 3D MSI dataset, described in Section 2.2 and Section 2.3 respectively.

## 2.2 Oral Squamous Cell Carcinoma 2D MSI Dataset

The first subject of our analysis is cancer located in the head and neck region (HNC). Head and neck cancers were the seventh most common cancers worldwide in 2018, with over 890,00 new cases and 450,000 deaths per annum [25, 34]. In 2020, the number of new cases raised to almost 932,000 and the number of deaths per annum to 467,000 [116]. It corresponds to 4.83% of all new cancer cases and 4.69% cancer deaths in 2020 worldwide.

The vast majority of head and neck cancers (>95%) are squamous cell carcinomas. They originate from stratified squamous epithelium lining mucosa of an organ like tongue, mouth, larynx, pharynx, salivary glands, and others. Despite significant improvements in treatment, oral squamous cell carcinoma (OSCC) have a low average survival rate compared to other types of cancers [113]. The major factors increasing OSCC risk have been linked to alcohol consumption, tobacco smoking, and HPV infection in the oropharyngeal region, while for lip cancer – ultraviolet radiation from sunlight exposure.

Today, the decision on the kind of treatment depends entirely on the tumor's location within the patient's body and the stage of its progression. However, as mentioned above, OSCC is a heterogeneous disease that may affect different organs, despite visual similarities observed upon an initial diagnosis or post-operational dissection. Uncompleted primary tumor resection generates a massive risk of local recurrence and a failure of the entire treatment.

Classical histopathological examination can miss out on sub-microscopic spots and yield uncertainty in interpretation. Analysis of molecular factors behind OSCC case, delineation of the tumor area, and the tissue of tumor origin could improve the overall prognosis for a patient through personalized therapy planning.

The dataset used in this work was first featured in [132]. For the tissue sample's biochemical preparation details, please refer to the original publication, as this is out of the scope of this work.

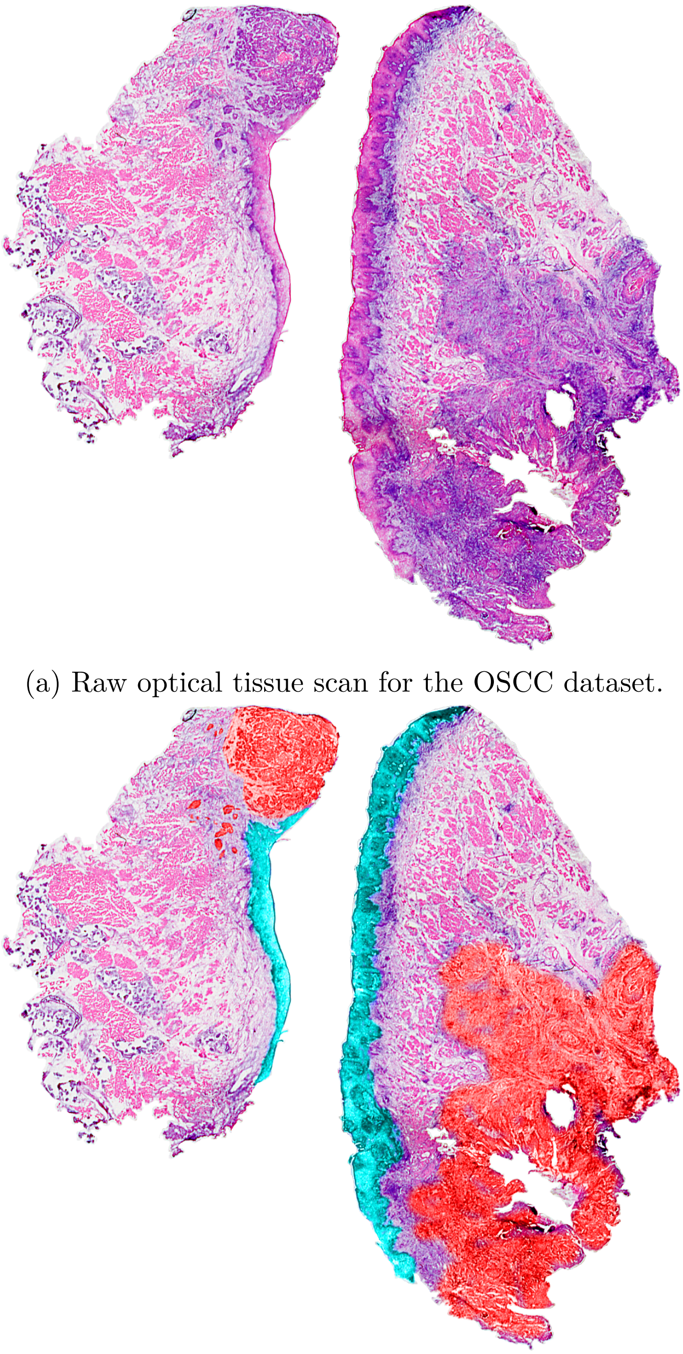
The biological material was collected from five patients who underwent surgery due to Oral Squamous Cell Carcinoma (OSCC). Tissue samples contained both tumor and surrounding healthy tissue.

Each specimen was cut into $10\mu m$ sections in a cryostat. During the sample preparation for the MS imaging, a high-resolution optical scan of each section was captured (see Figure 2.6a).

Tissue sections were subjected to peptide imaging with the use of a Matrix-Assisted Laser Desorption/Ionization Time-of-Flight mass spectrometer. Spectra were recorded within $m/z$ range of $800 - 4,000$. A raster width of $100\mu m$ was applied, and 400 shots were collected from each ablation point. The obtained dataset consisted of 45,738 raw spectra with 109,568 mass channels.

An experienced pathologist analyzed the optical scan obtained during the data acquisition process, and tissue regions were annotated (see Figure 2.6b). For the highest confidence of the results obtained in this work, we will focus on the two tissue samples out of the entire dataset (8,005 and 11,869 spectra), which have the highest confidence labels, as explained by the pathologist.

The preprocessing of the spectra was conducted in MATLAB. Standard preprocessing steps were applied to the spectra, following the process described in Section 2.1. Spectra were resampled to unify the $m/z$ axis across the dataset. Baseline was removed with MATLAB procedure `msbackadj()` from the Bioinformatics Toolbox. Peaks were aligned using Fast Fourier Transform-based spectral alignment [134]. The TIC normalization ensured a similar intensity level for all spectra. Finally, a GMM approach [95] was used to model the spectra. GMM locates the peak but also estimates the peak area



(a) Raw optical tissue scan for the OSCC dataset.

(b) The optical tissue scan annotated by an experienced pathologist. Red – tumor, cyan – healthy epithelium, magenta – other healthy tissue.

Figure 2.6: Optical scan of the tissue for the OSCC dataset.

instead of a raw magnitude provided by most methods. Note that the peaks in MSI spectra are right-skewed, so the neighboring GMM components resulting from that phenomenon were identified and merged to better correspond to actual chemical compounds. The resulting dataset is characterized by 3,714 GMM components corresponding to MSI spectrum peaks.

## 2.3 Mouse Kidney 3D MSI Dataset

This dataset was first featured in [87], released as a part of [88] and is reused in this work. For the tissue sample's biochemical preparation details, please refer to the original publication, as this is out of the scope of this work.

The biological material was collected from a wild-type mouse. The tissue sample contained the entire kidney of a mouse. Magnetic resonance imaging was conducted on the entire kidney within seven days after dissection, and the sample was further prepared for the MSI study.

The kidney was cut into 122 $3.5\mu m$-thick serial sections on a microtome. Tissue sections were subjected to Matrix-Assisted Laser Desorption/Ionization imaging. Spectra were recorded within $m/z$ range of $2,000 - 20,000Da$. A raster width of $50\mu m$ was applied, and 250 shots were collected from each ablation point. The obtained dataset consisted of 2,171,451 spectra with 7,680 mass channels each. The total size of such a dataset is 199 GB.

After MALDI imaging analysis, H&E-staining of sections was conducted; however, such data was not released publicly.

Finally, registration of slices is required to construct a 3D MALDI-imaging dataset from a set of numerous 2D serial sections. The registration was based on the optical images and an enhancement of the procedure described in [122]. Slices were translated and rotated first, but then an elastic registration was applied to correct local deformations of the slices, which occur during sectioning.

The original visualization of the registered dataset with annotated functional regions based on spectra clustering can be found in Figure 2.7.

Note, that the dataset released in [88] contains a subset of 75 consecutive slices (1,362,830 spectra). MRI and H&E images are unavailable.

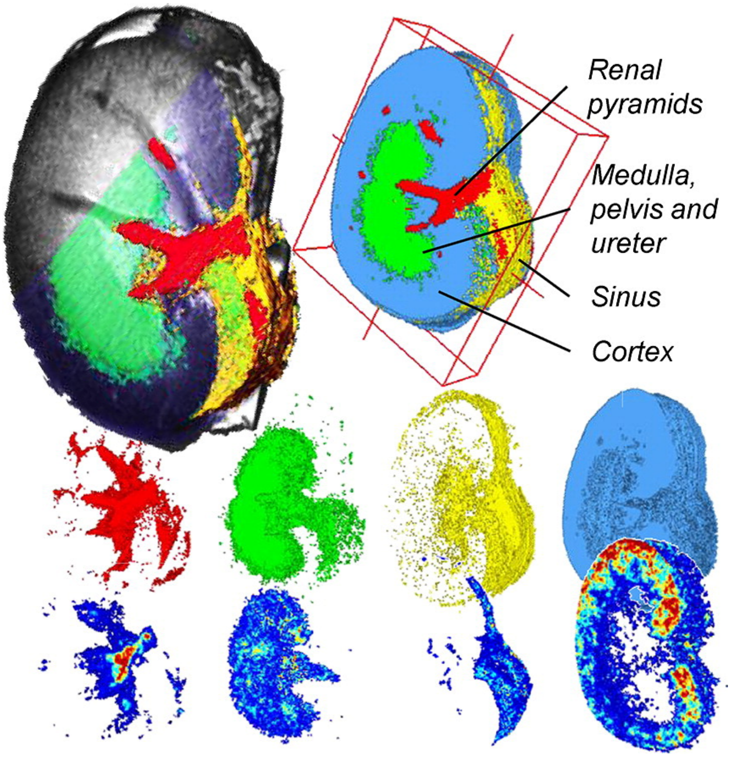


Figure 2.7: The original visualization of the mouse kidney 3D MSI dataset [87]. Registered MRI images were imposed with the clustering results based on the molecular information in the left upper part. Visualization of the clusters across the entire 3D volume in the right upper part. Two rows at the bottom feature the visualizations of the separate clusters (upper) and the concentration maps of correlated $m/z$ channels (lower).

Spectra smoothing was conducted with the Gaussian spectral smoothing with a width of 2 within 4 cycles. Top Hat algorithm was used for baseline reduction. Such prepared dataset was exposed in the `imzML` format.

Authors [88] clearly state that the purpose of this dataset is to benchmark an algorithm for MSI data processing under the condition of high-volume real data. No annotations are available, MRI and optical images have not been released. Thus the visual comparison of the obtained tissue structure is the only possible assessment. We will use this dataset to test the scalability of algorithms and whether the obtained result could be associated with the structure presented in Figure 2.7. Limited assessment opportunities seem consistent with a recent remark that an appropriate niche for 3D MSI is yet to be found. However, the existing 3D datasets are challenging representatives of high-scale molecular data, perfect for benchmarks [9].

# Chapter 3

# Methodology of Big -omics Data Clustering

Efficient analysis of thousands of Mass Spectrometry Imaging spectra requires adjustments to the feature space, regardless of the slight differences in spectra preprocessing pipeline (see Chapter 2) [66, 117, 129, 128]. Despite some variability in the process, the preprocessing pipelines address the same physical phenomena and lead to a similar data representation: the number of ions observed in a specific area of the investigated tissue.

## 3.1 Clustering Algorithms and Feature Engineering Methods

As we focus specifically on unsupervised knowledge discovery, only fully unsupervised methods are within the area of interest of this work. Unsupervised feature engineering methods can be divided into the following groups [97, 14]:

- Filtering methods – they eliminate features based on some (usually fixed) threshold, taking into account feature average value, signal-to-noise ratio, variance, or other feature parameters.

- Linear feature transformations – they estimate a new set of features as a linear combination of the original features so that the reconstruction error is the smallest. Notable methods of this kind are Principal Components Analysis and Linear Discriminant Analysis.

- Non-linear feature transformations – they estimate a new set of features as a non-linear transformation of the original features. Most often, the goal of such transformation is to achieve the smallest error on pairwise dissimilarities between observations. Reconstruction capabilities are not always guaranteed. Notable methods of this kind are Uniform Manifold Approximation and Projection, t-Shaped Stochastic Neighbor Embedding, and Non-Negative Matrix Factorization.
- Deep learning-based approaches – these are further discussed in the Section 3.5, as they constitute an entirely new group of methods and a new trend in big -omics data clustering.

Note that all these kinds of methods are applied to the dataset once and globally, thus discarding nuances that could be relevant for hierarchical analyzes.

Similarly, clustering methods also are divided into a few groups [44, 54, 104]:

- Centroid-based – a representative of each group is selected, and redefinition of groups occurs for these representatives. Examples of such algorithms are K-Means or K-Medoids (using *medoid* as a representative instead of mean-based centroid).
- Connectivity-based – a rule is defined to select the next clusters to merge. The most recognized example is hierarchical clustering.
- Distribution-based – a predefined distribution model is fit to the data. One of the most popular is the expectation-maximization method fitting a Gaussian Mixture Model to the data.
- Density-based – clusters are defined as the areas with a higher density of points than the remaining dataset. They often introduce a notion

of outliers – points that are located in low-density areas. DBSCAN is an example of such an algorithm and requires an upfront definition of expected density.

### 3.1.1 High-Dimensional Data Clustering

#### Subspace Clustering

In the domain of high-dimensional data clustering, combining a feature filtering method and clustering algorithm is a special case of *subspace clustering*. This combination would be one of the simplest yet efficient methods widely used today. However, there exist more sophisticated methods.

One could bring the example of SUBCLU [67]. SUBCLU is an algorithm that joins DBSCAN with a kind of forward feature selection. First, it starts with a set of single-feature subspaces. DBSCAN partitions the data in such subspaces, and candidate new features are added from subspaces that differ by just a single feature. In such an extended subspace, SUBCLU identifies clusters again, operating in the regions discovered in a less-dimensional subspace. Unfortunately, the DBSCAN algorithm requires numerous ranged queries to find a neighborhood. These are usually realized with a supporting structure called $k$-d tree to avoid an exhaustive search when processing an observation. As the subspace for DBSCAN clustering changes, the supporting $k$-d tree needs to be recomputed. Such a recomputation vastly increases computational complexity for datasets with the number of features of similar magnitude to the number of observations (or higher), which is often the case for biological data.

Another approach, CLIQUE [6], identifies high-density subspace regions starting from a set of histograms in one-dimensional space. CLIQUE creates candidate subspaces via merging subspaces that differ by a single dimension, similarly to SUBCLU. Authors prove that a cluster in $k$ dimensions must also constitute a cluster in $k - 1$ dimensions. Hence, a candidate $k$-dimensional subspace is discarded if it has a $(k - 1)$-dimensional subspace that was not found dense previously. To further limit the number of candidate subspaces, CLIQUE discards dense subregions with low coverage over the input dataset.

In the next step, CLIQUE identifies which dense regions were spatially adjacent and merges them to form the clusters. Finally, the authors describe how to obtain an expression that precisely describes the cluster limits. Included experimental evaluation shows linear scalability with the number of observations but much worse with the number of dimensions (considered up to 100 dimensions). Additionally, increasing the target dimensionality prolongs the experiment time non-linearly (the authors consider up to ten dimensions for cluster construction).

In contrast to SUBCLU and CLIQUE, *intelligent Minkowski metric Weighted K-Means* (iMWK-Means) [37], is a representative of *soft subspace clustering algorithms*. It is basically a K-Means algorithm with few modifications. First, it uses weights to express the importance of dimensions in the input dataset. As the weight assignment is a non-trivial task, the authors discuss a few variants, depending on the remaining modifications of the K-Means algorithm. Secondly, the number of clusters and initial centroids are obtained with so-called *Anomalous Clustering* [33]. Finally, the distance is captured with the Minkowski metric, a known generalization of the Euclidean metric, which could boost the effect of weighting in prioritizing the feature importance. Authors outline that the mentioned set of modifications highly influences the running time for K-Means, as cluster center recomputation requires a more complicated process, more similar to K-Medoids [19] than classical K-Means. The presented results indicate that iMWK-Means can achieve great accuracy on the evaluated datasets. However, it requires tuning of the Minkowski metric exponent, which may not be straightforward to calibrate in an unsupervised setup, as it significantly varied between conducted experiments.

EBK-Modes [29] uses feature weighting and a well-known algorithm – K-Modes, similarly to what iMWK-Means uses. The authors propose an entropy-based approach for measuring the relevance of each categorical attribute in discovered clusters. Although biological big data rarely is categorical, EBK-Modes is worth mentioning due to its unsupervised assessment of the feature importance. Experimental evaluation exhibits increased accuracy, adjusted Rand index, and F-score obtained with EBK-Modes.

The newest methods for subspace clustering introduce neural networks, like *Graph Regularized Residual Subspace Clustering Network* (GR-RSCNet) [28]. The authors proposed GR-RSCNet for the clustering of hyperspectral images. At its core, one may find a convolutional autoencoder with residual connections to improve the flow of the gradients. Training of the neural network enforces sparse representation of the affinity matrix obtained for the latent space as a part of the used loss function. Moreover, the affinity matrix is regularized by the spatial coordinates between the hyperspectral image pixels. Finally, the affinity matrix is segmented using spectral clustering to generate the final clusters. Results attached by the authors indicate a significant increase in accuracy and normalized mutual information, although the comparison occurs mainly with more limited variants of the same method.

### Projected Clustering

Another group of methods for high-dimensional data clustering is called *projected clustering*. It operates with a classical clustering algorithm and a custom metric through which clusters may be formed in different subspaces. Such a formulation poses a severe constraint on which clustering algorithms could be used. One of the most popular would be the K-Medoids algorithm that minimizes the intra-cluster sum of distances. It could be combined with distance measures like Gower distance, which is often applied to compare the responses in the numeric and non-numeric data simultaneously, and also with some parts of data missing.

The primary benefit of such an approach is explained in [5], where authors indicate that traditional feature selection algorithms pick specific dimensions before clustering and lead to a significant information loss. The featured algorithm PROCLUS requires the number of clusters $k$ as an input and the average number of dimensions in which a cluster should be defined. It estimates $k$ clusters (plus a set of outliers) and $k$ sets of dimensions, which were used to obtain the related clusters. The idea behind the algorithm is to find a set of medoids based on Manhattan segmental distance, which, similarly to Gower distance, is relative to the current dimensionality of the dataset. First, PROCLUS finds a superset of the medoids with many outliers, and then the set gets refined to obtain a robust set of clusters.

There are many issues characteristic of the projected clustering: a non-determinate result, dependence on the order of processing, low robustness against noise, multiple scans over the entire database, and non-linearity in the number of dimensions. Authors of the PreDeCon (subspace PREference weighted DEnsity CONnected clustering) approach [20] claim to address these issues. The algorithm captures local directions of the increased density in the data and extends classical DBSCAN for density-based clustering. It uses a weighted Euclidean distance to compute smaller and more specific clusters instead of trying to cluster all available points in the dataset. The user selects the parameter $\lambda$ specifying the threshold of local dimensionality for the distance measure, in addition to the $\epsilon$ specifying the threshold of distance. The PreDeCon method complexity is quadratic in the number of observations.

A more recent method [140] uses a fuzzy K-Meansalgorithm with a flexible manifold. Authors propose two algorithms: fuzzy K-Means with pattern shrinking and projected fuzzy K-Means with pattern shrinking. The first introduces the concept of joint clustering and pattern shrinking, which denoises the original dataset based on currently known cluster membership information during each iteration. The projected alternative additionally compresses the dimensions irrelevant for currently known cluster membership. Authors validate both methods via clustering RGB images. Although interesting, the practical applications of the projected fuzzy K-Means with pattern shrinking are severely limited due to high computational complexity – cubic in both the number of observations and the number of dimensions.

Projected clustering seems much less popular nowadays, as its computational complexity is high, and the interpretation of clusters is complex (or even impossible) due to unclear relations between input features and the obtained clusters.

### Projection-Based Clustering

A linear or non-linear feature transformation combined with a clustering algorithm is called *projection-based clustering* in the domain of high-dimensional data clustering. This area covers all the approaches like Principal Components Analysis, t-Shaped Stochastic Neighbor Embedding [124], Uniform Manifold Approximation and Projection[82], and others, followed by a clustering method of choice. The data is first reduced to a low dimensionality projection, and then the projected observations are clustered with a classical clustering method for low dimensionality.

This approach is probably the most popular, as it suffices that advancements occur in the area of projection methods, not necessarily the clustering algorithms. At the same time, it has a severe limitation of significant information loss, regardless of the projection method [5].

## 3.2 Clustering Quality

Clustering quality can often be assessed with metrics like Dunn's index [47], or GAP statistic [119] in terms of cluster separability, and Adjusted Rand Index [72] in terms of label relevance (if labels are available). Although they were introduced for low-dimensional data, there are studies [74, 102] that compare their usefulness for high-dimensional data and subspace clustering. Unfortunately, these studies consider hundreds of features a high-dimensionality problem, while it is still at least an order of magnitude less severe than for Mass Spectrometry Imaging data clustering. Hence, we will focus on domain-related work [128].

## 3.3 Classical Approaches to MSI Data Clustering

In this section, we will focus on the approaches which do not leverage any aspects characteristic of MSI data despite that provided biologically relevant results. A careful reader may realize that most of these methods are, in fact, projection-based clustering, which imposes substantial limitations on observed the level of detail in the results.

The first MSI computational attempts relied primarily on linear dimensionality reduction like Principal Components Analysis and simple clustering algorithms like hierarchical clustering or K-Means. The study [39] evaluates the usefulness of automated analyses of the entire dataset at once, as opposed to the previous manual analyses of single selected ion maps or individual spectra. Authors select ions correlated with the visual composition of the tissue in the optical scan. Corresponding ion maps are used as red, green, and blue channels of a reference image. This reference image is subsequently compared to the spatial distribution of the three top components obtained via PCA. Authors argue that the differences between histology and PCA scores may be related to inhomogeneous sample preparation but also the presence of molecular species which do not follow the histological features. Independently, hierarchical clustering is conducted on a PCA representation reduced to 70% explained variance. The clustering happens in an unsupervised way; however, the number of clusters is semi-supervised. The threshold for the number of clusters is selected arbitrarily to match the histology.

[65] provides a *user guide* to MSI data analysis. Authors describe much more comprehensive data preprocessing methods than used in most studies, including spatial smoothing methods from the image processing domain and others. In terms of feature extraction, first, a strong limit on the number of considered mass channels is imposed, and only then dimensionality reduction method is applied. Therefore, the PCA or Non-Negative Matrix Factorization (NNMF) they mention is used with up to a few dozens of mass channels. In terms of histology-independent analysis, which is the subject of our study, the authors indicate the usefulness of K-Means clustering and hierarchical clustering.

Another study considers two datasets of spatial dimensions of 103x169 and 104x168 respectively and 8,000 $m/z$ values each [106]. A matrix of experiments is conducted, with PCA dimensionality reduction and without, and five configurations of clustering methods. PCA was set to explain 90% of the variance. The configurations of the clustering methods were: Fuzzy

K-Means and K-Means with four different distance measures (Euclidean, Manhattan, correlation, and cosine). Both clustering methods are used with nine different values of the number of clusters $k \in [2, 10]$. The results from the matrix of experiments were evaluated against the Calinski-Harabasz index and manual annotations. Annotations were automatically constructed from thresholding some preselected ion images, and the correlation between thresholded images and the obtained clusters is compared. Authors indicate that the Euclidean and correlation distance reveal finer structural details than the Manhattan and cosine distance. An interesting observation made in the publication is that configurations with Euclidean distance and other distance measures tend to correlate highly with disjoint sets of $m/z$ images. Thus it may be valuable to identify distance metrics that yield complementary results.

More recent work introduced a two-phase graph-based algorithm that optimizes computer memory utilization [41]. It underlines the fact that K-Means was sufficient for segmenting the matrix from the tissue or highly differentiated tissues. However, it may provide unsatisfactory results for a larger number of fairly similar anatomies. In response to that issue, the authors propose using a more sophisticated clustering algorithm, which in turn comes with an increased computational cost. The suggested solution is to efficiently sample the dataset so that the batches of data analyzed at once do not incur a significant penalty, with the benefit of potentially increased clustering accuracy. First, the subsets are clustered independently, and then cluster representatives from the compression set. Then, the compression set is clustered to define a global partition for the entire original dataset. The core of their algorithm is graph-based clustering, namely spectral clustering, as it is a very effective method for classical image clustering. The method is validated with real and synthetic data of the size up to 300,000 spectra and subset sizes ranging from 17,000 to 25,000 spectra. The clustering was conducted on the set of approximately 2% smallest eigenvectors, both in the sample and in the compression set. In the results, the authors present that the graph-cuts algorithm revealed a much more detailed molecular structure of the tissue than the basic K-Means approach.

The subject of non-linear dimensionality reduction for MSI data was tackled by [3]. Authors proposed to use the Hierarchical Stochastic Neighbor Embedding (HSNE) [93], an interactive extension of the popular visualization method t-Shaped Stochastic Neighbor Embedding (t-SNE) [124]. The original t-SNE method scales quadratically with the number of observations. Therefore, it cannot be applied for a large-scale MSI dataset analysis. HSNE is based on the concept *Overview-First, Details-on-Demand*. On a large scale, the embedding shows dominant data structures (an *overview*). Then, it uses landmarks from high-level structures to compute hierarchical local embeddings, which refine the visualized information (the *details*). Such an approach keeps the memory and computational complexity under control, even for massive 3D MSI datasets. The HSNE method was benchmarked on the mouse kidney 3D dataset [88] and allowed to identify the major functional structures in the 3D volume. Note, however, that this approach is semi-supervised and requires the user to manually digest the obtained embedding to define regions of interest for a drill-down. A significant subset of the original dataset was discarded in the visualization.

Both t-SNE and HSNE lack an essential property: it is impossible to embed new data to a computed embedding with these methods. Uniform Manifold Approximation and Projection (UMAP) algorithm was derived [82] to respond to that need. One of its most relevant input parameters is the distance measure. As we know from other studies, there was no clear distinction on which distance would yield the best results consistently. Fortunately, this was evaluated in the context of UMAP [114]. The authors assessed a few distance measures: Euclidean, correlation, cosine, and Chebyshev distance. Part of the evaluation is conducted with spatial autocorrelation – spatially neighboring pixels are expected to embed into a similar area of the latent space. Finally, the authors showed that UMAP yielded superior runtimes compared to t-SNE and that correlation and cosine distances achieve the best results for MSI data.

## 3.4 Approaches Considering Spatial Information

One of the first clustering approaches considering spatial information was [10]. The authors explained that denoising each individual $m/z$ image would be more natural than post-processing the resulting classification maps. However, the challenge was to propose a denoising method that would not erode the molecular details at the boundary of two neighboring morphological regions. Therefore, they suggested using an edge-preserving Chambolle algorithm [32] with a Grasmair modification [53] that locally adjusts the denoising scale. First, the peak picking was applied out of 5027 mass channels selected 110 peaks. Then, the ion map for each selected peak was smoothed with the Chambolle algorithm. Finally, a high dimensional discriminant clustering [23] was applied to obtain the partition of the dataset. They recommended selecting the number of clusters manually, based on histology examination. Upon visual inspection, the proposed method seemed to reveal structural details consistent with the schematic of the anatomical structure of the rat brain. The comparison with smoothing methods discarding information about the image edges showed the superiority of the approach. Similar outcomes were obtained with another dataset of the human brain.

The next step in the edge-preserving denoising was presented in [12]. Two clustering methods were presented, differing in the smoothing method: spatially aware clustering and spatially aware structure-adaptive clustering. Both start with the ion images denoising. Spatially aware clustering averages the ion map values based on Gaussian weights defined using the spatial distance between spectra. Spatially aware structure-adaptive clustering uses a form of a bilateral filter, which compares spectra similarity based on the molecular information but also includes a Gaussian weight dependent on the spatial distance. To ensure the low computational complexity of both methods, the authors proposed to project the spectra using the FastMap algorithm and reduce the dimensionality of the dataset simultaneously. After the embedding, they applied classical K-Means clustering. With the increasing radius of the smoothing method, resulting segmentation maps were more spatially consistent, with less noise in the labels. Again, the evaluation was conducted upon visual comparison with the rat brain atlas. Although, the authors were able to identify tissue slice preparation defects and the artifact caused by the non-edge-preserving spatially aware clustering. Once again, a separate evaluation was conducted on an independent human tumor dataset and revealed the molecular composition of the tissue.

An even more challenging area was tackled in [122] for 3D data. In contrast to 2D MSI datasets, the registration of consecutive tissue sections does not align the raster perfectly with each other. Moreover, the slice thickness differs from the raster width. Hence, one cannot consider the spectrum a square in a grid anymore but rather a point in a 3D point cloud. The authors used the nearest neighbor search in a 3D space to generalize the Chambolle algorithm from the usual grayscaled 2D raster. To scale well for 3D data, bisecting K-Means was selected, which separated data until singleton clusters were obtained. It generates a binary tree of the clusters but at the same time does not require storing the pairwise distances between all the points. The K-Means algorithm was used with correlation distance. Authors also considered Manhattan and Euclidean distance, as well as agglomerative hierarchical clustering with the average linkage and the correlation distance. Algorithms were first evaluated with low-scale 2D datasets and confirmed high reproducibility of histological and chemical spatial structure. For the 3D data, the segmentation was conducted to separate the spectra into three segments, and correlated ion maps were investigated to confirm the relevance of the results.

The following approach focuses much less on the individual spectra but rather on ion maps representation of the MSI dataset [11]. The spatial segmentation discussed up to this point answered the question: *which pixels in the MSI dataset do have similar mass spectra?* In the publication, the authors redirect the focus onto another question: *what do all m/z images of MSI dataset look like?* Therefore, the PCA decomposition presented in the cited work does not represent a spectrum anymore but an ion image. Groups of similar ion images were identified using the Gaussian Mixture Model. The results were validated by analyzing the cluster distributions in the PCA decomposition (clusters obtained without embedding). Secondly, the intracluster variance was used to estimate the compactness of the clusters. These two validation approaches were summarized into a step-by-step instruction for a human-in-the-loop approach, which should allow researchers less knowledgeable about machine learning to take advantage of the method. The process was shown on the example of two datasets: rat brain coronal section (compared to the rat brain atlas) and human larynx carcinoma sample (compared to histological images). Benefits of such an approach include the detection of artifacts missed in the data preprocessing and data quality assurance.

The EXIMS method [133] addresses another area of exploiting spatial information: estimating the biological relevance of the observed ion maps. Authors categorized multiple $m/z$ images into unstructured and structured, with a clear distinction between four kinds of structures: regions, curves, gradients, and islets. The following steps were proposed to leverage information about potential patterns:

- First, apply median filtering with a 3x3 neighborhood to reduce the technology-driven artifacts.
- Enhance the contrast of the grayscale ion map using histogram equalization – this allows to reduce the number of hot spots in the $m/z$ images.
- Quantize ion map into eight levels of gray.
- Compute gray-level co-occurrence matrix.
- Multiply obtained co-occurrences matrix with a predefined weights matrix.

The predefined weights appeared to be the most successful for structures categorized as regions and curves. The algorithm's running time highly depends on the bin size applied on the spectrum.

The recent work in spatial clustering featured the community detection method GRINE (analysis of GRaph mapped Image data NEtworks) [135]. It focused on grouping the molecules into communities. First, the interesting

$m/z$ values needed to be preselected from the entire dataset. Then, an affinity matrix was computed using the Pearson correlation coefficient for a reduced set of ion images. The matrix was transformed into an adjacency matrix, and the connections were automatically pruned. Communities were detected in a divisive manner with the leading eigenvector method. The authors present the software implementation and the results based on simulated Gaussian peaks and two MSI datasets with preselected peaks. Unfortunately, despite the well-maintained open source implementation, we could not run the analysis for the OSCC dataset introduced in Section 2.2. Authors suggested massive manual preselection of peaks before applying this method.

## 3.5 Deep Learning Approaches

The first deep learning approach to MSI was presented in [117]. Autoencoder applied there is an alternative to Non-Negative Matrix Factorization and PCA, which allows for the extraction of latent ion images with highly detailed molecular structures. Although [114] indicates the benefit of speed in UMAP compared to training an autoencoder, this work initiated a series of auspicious deep learning efforts in the MSI field. The method is discussed further in Section 5.2, after a brief introduction to variational autoencoders in Chapter 5.

In Section 3.3 we discussed the limitations of t-SNE and HSNE, however as has been pointed out [61], both these methods relied on the most basic definition of t-SNE, while there was also available a parametric t-SNE [123] alternative by the same author. It used a Restricted Boltzmann Machine (RBM) to achieve the same results as t-SNE, with an additional benefit of increased scalability. Neural network architecture is proposed that consists of RBM blocks. They are first pretrained independently, then merged into a single feed-forward network and fine-tuned, using the same Kullback-Leibler divergence as the non-parametric t-SNE. Such an approach had two benefits: increased stability of the results and an opportunity to embed new data as they are captured. Additionally, the authors brought up an important aspect of comparison of the parametric t-SNE to the competing autoencoders. There are a few differences worth considering when deciding between these methods:

- They optimize different cost functions. Parametric t-SNE aims to preserve local neighborhoods, while autoencoders aim to minimize the original data reconstruction error. These two goals may be correlated to some extent but are not the same.

- Parametric t-SNE has a benefit of an architecture smaller by half, as it does not require the decoding part – it is just an encoder. Therefore it is more shallow, which can train faster and consume fewer resources.

- For a latent space not large enough to accommodate all properties of the data (e.g., 2D space, which is convenient for visualization), parametric t-SNE pushes the natural clusters in the data apart, which opposes the embeddings provided by autoencoders, in which these natural clusters partially overlap.

The authors concluded by comparing parametric t-SNE to an autoencoder of a corresponding architecture: the exact size of the encoding part as the parametric t-SNE and a mirrored architecture for a decoder.

Although we mainly discussed the unsupervised approaches up to this point, the deep learning section will also tackle supervised learning, as the architectural ideas often can easily be transferred between the methods. The first example could be the application of convolutional neural networks (CNN) to MSI data classification [17]. MSI data have the property that consecutive mass channels in the spectrum should be correlated to some extent, e.g., a ToF peak is spread over several $m/z$ bins. Therefore they seem a good target for applying a convolutional neural network directly on the raw spectrum (without peak picking). Conceptually, the first layers could learn peaks. However, the authors argued that the mid-level features could represent isotope patterns from the data or adduct patterns of the same peptide. Furthermore, the highest-level features could recognize tryptic-digested proteins that contribute to patterns across the entire mass range. On the architecture side, the authors applied convolutional transforms together with a locally connected layer that has *unshared weights* and allows for learning local characteristics of the spectrum across the entire $m/z$ range. Additionally, residual connections were introduced due to the inspiration of deep Residual Network architecture. The method was compared to the IsotopeNet, which outperforms it in classification metrics. However, the authors discussed that a closer analysis revealed biological connections to known biomarkers for the discrimination of adenocarcinoma and OSCC.

The idea of a convolutional neural network for classifying MSI data was further developed with atrous convolutions [125]. Authors indicated that cancer biomarkers might be more spread over the $m/z$ axis. Thus it could be beneficial to increase the receptive field of the neural network without additionally increasing neural network depth. The dilation rate was increased gradually with the depth of the network to prevent loss of the resolution of the input signal. The architecture of the neural network used in the study is based on the IsotopeNet. However, it replaces classical convolutional layers with atrous convolutions. The method is validated with two datasets: lung tissues sample from 12 patients representing two subtypes of lung cancer and bladder tissue samples from 9 patients representing healthy and cancerous tissue. With these datasets, the authors demonstrated an improvement in balanced classification accuracy and F1 score.

For another kind of MSI (ToF-SIMS), a method based on self-organizing maps was applied [52]. Output layer neurons were assigned a position on a toroidal map, and a 2D color scale was created. As the self-organizing map learned the data structure, each mass spectrum converged to a position on the toroid based on the Euclidean distance between the neuron output and the original spectrum values. The results were used to visualize the molecular differences imposed on the histological images. Such visualization allowed straightforward visual identification of an ROI in the collected data.

Supervised deep learning methods require high-quality annotations, which are obtained in a time-consuming process engaging a medical expert. Thus, they are often unavailable, especially when a sub-tissue-level classification is required. Multiple instance learning was proposed [55] to address the issue. The main difference was that instance-level (spectrum-level) predictions were often avoided. Instead, a bag-level (e.g., tissue-level) prediction was made for a set of spectra. The authors built the method based on mi-SVM but replaced the Support Vector Machine with a convolutional neural network. The method was evaluated with simulated and real-life datasets and showed superior performance compared to the mi-SVM and classical instance-level predictors (SVM and CNN). The power of weak supervision was demonstrated: a dataset without a complete annotation was exercised, and it was easy to propagate the assumptions about the annotations into the training pipeline.

The similar issue is further addressed by *cumulative learning* [110]. Authors argue that transfer learning is not entirely applicable to MSI datasets, as there are major differences between a usual transfer learning scenario and the limitations of using MSI data in biomedical research. Models dedicated to transfer learning were trained to generalize for multiple (even thousands) classes and sometimes even for multiple tasks. At the same time, MSI's reality is that there are up to a few classes and a single task. The amount of data is severely limited, which also impacts the variability to which a model could be exposed. Therefore, the authors train the model for several tasks to converge to an optimal model. Besides the classification, the model learns cross-instrument representations and addresses the fluctuations in the instrument's performance. Authors compare the performance of CNN trained from scratch, transfer learning, and cumulative learning approaches. The non-reproducible technical factors got filtered by the CNN and increased the robustness of molecular pattern recognition. Cumulative CNN offered a unified solution very robust towards the different sources of variance.

The other work builds on the success of UMAP and t-SNE as universal dimensionality reduction methods [42]. Unlike most approaches, it does not train the neural network with the cost function specified similarly to UMAP or t-SNE. However, it learns the dimensionality reduction in a black-box manner. The complexity of dataset embedding can easily be controlled at the cost of including the nuances in the final embedding. However, there are two more advantages: dimensionality reduction can be applied to unseen data in an online manner, and a reverse transformation can be learned, which is impossible for classical t-SNE.

One of the most recent spectrum-oriented deep learning approaches was based on a Variational Autoencoder [2]. Authors evaluated the method with the mouse kidney 3D dataset [88]. Precise details of the algorithm are featured in Chapter 5.

The subsequent work introduced a transfer learning approach, benefiting from well-known computer vision datasets like ImageNet [139]. Authors suggest that the processing pipeline mimics a human reviewing manually the ion images and looking for similar patterns. Therefore, a pretrained Xception neural network was used to create embedding of the ion map patches. Patches were constrained with a real-life size of $1 - 2mm$ and the possible input size to the Xception network. Too small patches needed to be upsampled prior to the embedding. Max pooling of the patch-level embedding provided the embedding for an entire $m/z$ image. Ion maps of a similar embedding are clustered together and merged into one. Further analysis was conducted with such *neural ion images* after the deduplication of the molecular information.

## 3.6 Summary

Analysis of multiple approaches, both general and dedicated to Mass Spectrometry Imaging data processing, leads to a surprising observation: the vast majority of methods are based on the assumption that the descriptor importance remains unchanged for the entire dataset. However, most of the experimental setups focus on the discovery of the differences inter- and intracluster at the same time, combining some of the following sample questions (and other questions not mentioned here) into a single study:

- Are the tissue type differences dominating over inter-patient differences or vice versa?
- What are the molecularly heterogeneous regions of the tissue?
- Is there any correlation between molecular biomarkers and the pathologist delineation of the tumor?
- What are the differences between a tumor and healthy tissue?

- Are there molecularly diverse subtypes of a tumor that differ in their origin and preferred therapy?

- What are the differences between a tumor of one type and a tumor of another type?

Reliable answers to even just some of these questions simultaneously cannot be obtained with Principal Components Analysis, UMAP, or other one-step methods, as the projection-based clustering approach *by definition* discards the non-dominating variability.

# Chapter 4

# Divisive Intelligent K-Means

## 4.1 Framework idea

The primary concern of the Divisive Intelligent K-Means (DiviK) framework is to limit the negative impact caused by information loss when using one-step methods. Therefore, it conducts the clustering procedure hierarchically and adjusts the feature space locally for each discovered subregion. Each subsequent local feature space optimization starts from the entire set of features so that the original information is available regardless of the current depth of the analysis. The idea is presented in Figure 4.1.

## 4.2 Methods

In this section, we will discuss three major components of the Divisive Intelligent K-Means framework:

- feature engineering method;
- clustering method;
- stop condition.

The selection of the algorithms was inspired by the state of the art methods but with a strong emphasis on model simplicity and explainability for the

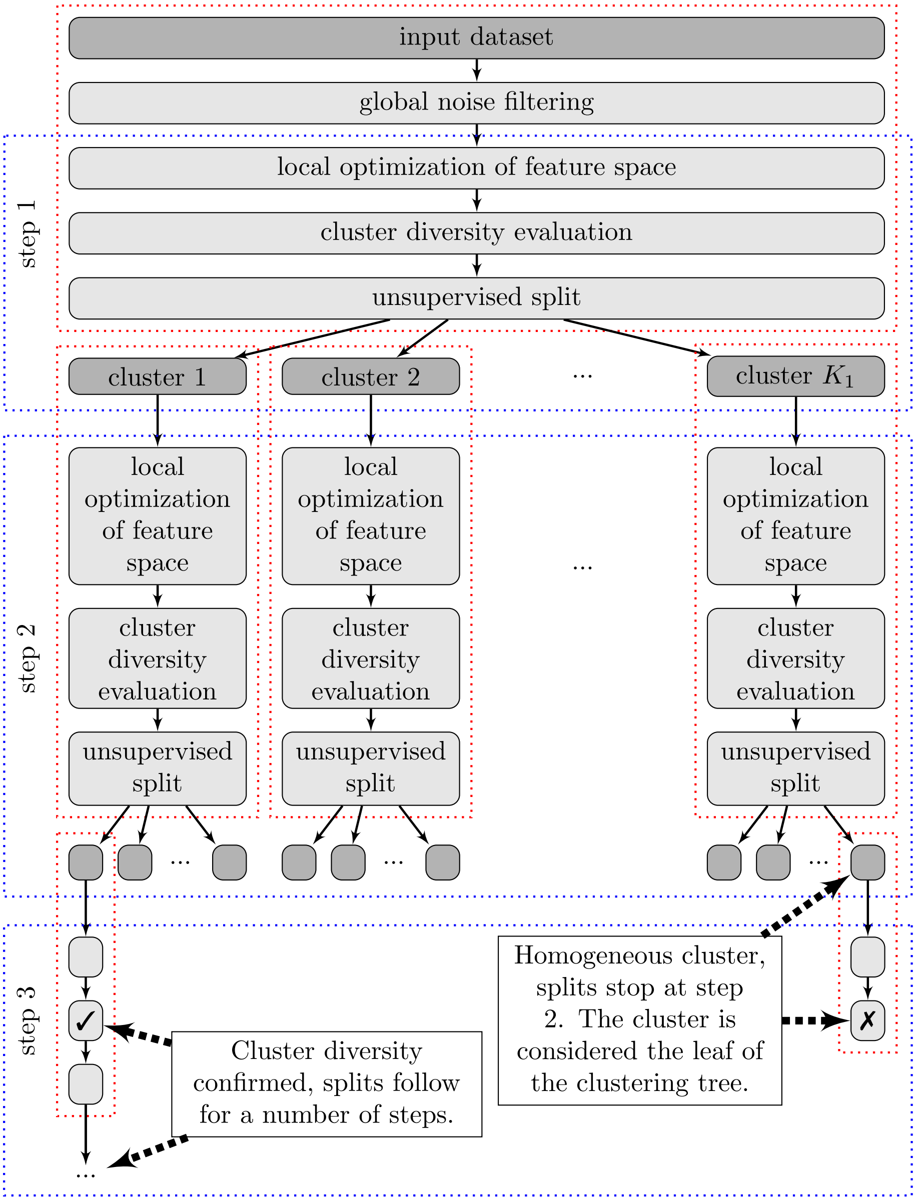

input dataset
global noise filtering
local optimization of feature space
cluster diversity evaluation
unsupervised split
step 1
cluster 1
cluster 2
...
cluster $K_1$
step 2
local optimization of feature space
cluster diversity evaluation
unsupervised split
step 3
✓
✗
Cluster diversity confirmed, splits follow for a number of steps.
Homogeneous cluster, splits stop at step 2. The cluster is considered the leaf of the clustering tree.

Figure 4.1: Flow diagram of the Divisive Intelligent K-Means algorithm.

evaluation process. All three elements could be further modified or entirely replaced without abuse of the initial Divisive Intelligent K-Means framework concept to adjust the method for another kind of data or even another domain.

### 4.2.1 Feature Selection Through Intelligent Filtering

As described in Chapter 3, there are many approaches for feature engineering in high-throughput biological data, which could be roughly separated into the following buckets:

- filtering methods – like fixed threshold filtering;
- linear transformations – like Principal Components Analysisor Linear Discriminant Analysis;
- non-linear transformations – like Uniform Manifold Approximation and Projection or Non-Negative Matrix Factorization;
- deep learning methods – like Variational Autoencoders.

Each item on the above list is slightly more sophisticated than the preceding one and usually provides a more comprehensive insight into the actual data structure. Despite that fact, all these methods discard nuances by design.

To demonstrate the potential of the Divisive Intelligent K-Means framework, we will start with the most basic group of methods – filtering. We will show that the local optimization of the feature space can provide a benefit over sophisticated scenarios applied once and dataset-wise, even when combined with a relatively simple feature engineering method.

Numerous filtering methods are applicable for high-throughput biological data [126, 136, 103, 80, 81, 60, 133]. Most of them are general enough to be used with any tabular data. However, the most sophisticated filtering procedures like [60, 133] rely intensively on spatial patterns in the Mass Spectrometry Imaging data and related domain knowledge. Unfortunately, the spatial patterns may not be preserved for the subregions obtained in the clustering process, as the subregion may be spatially discontinuous and scattered.

Due to promising results of previous studies [94, 132], we will continue with Gaussian Mixture Model (GMM) based method [80] to demonstrate the benefits of local feature engineering.

The careful reader may notice that Figure 4.1 contains two kinds of blocks influencing features: global noise filtering and local optimization of feature space. Both processes follow a similar schema, presented in Figure 4.2.

For dataset-wide one-time global noise filtering, we use an average abundance of the feature (top left panel of the Figure 4.2). We decompose the histogram of the average feature abundance into a GMM, with the number of components optimal in terms of the Bayesian Information Criterion (middle left panel). For each observed value of the average abundance, we calculate the conditional probability for each Gaussian component. Then we apply the maximum classification rule, which leads to the interpretation that the crossing points of the neighboring GMM components become filtering thresholds. For the average abundance, we discard the peaks represented by the first non-artificial GMM component as it is most likely noise-related (bottom left panel). This procedure is conducted just a single time on the entire dataset.

For local optimization of the feature space, we use ion abundance variance (top right panel of the Figure 4.2). The procedure follows similarly, but for the abundance variance, we persist only the peaks represented by the topmost GMM components, not less than 1% of all the peaks.

In-depth implementation details of the GMM-based filtering can be found in the related literature [80, 95, 96, 81].

### 4.2.2 K-Means

K-Means problem [78] is a problem of partitioning $n$ observations into $k$ clusters, in which each observation belongs to the cluster with the nearest centroid – a representative of a cluster, defined as the mean of the observations inside.

The globally optimal solution to K-Means problem partitions observations in such a way that the sum of squared distances between respective centroids and cluster members across all clusters is the lowest. However, solving the

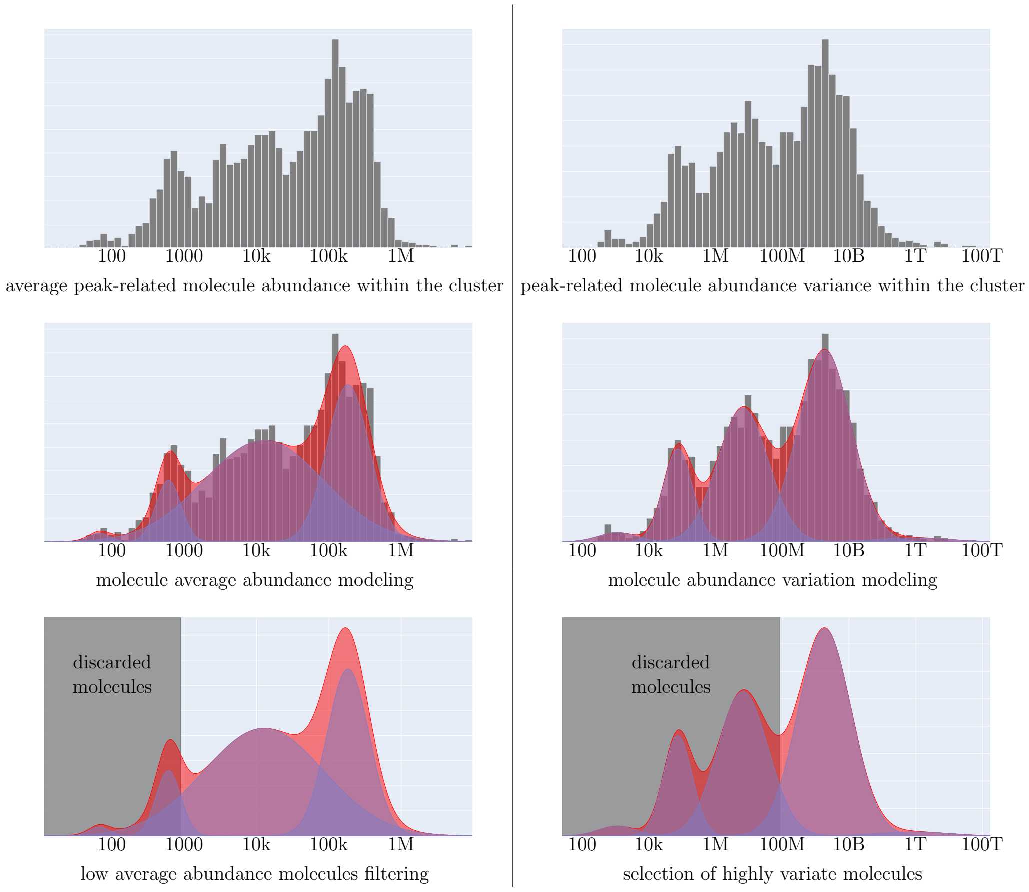


Figure 4.2: Visualization of the GMM-based feature filtering procedure.

problem for global optimum is computationally difficult (NP-hard). Therefore, we use a popular Lloyd's algorithm [76], which is an efficient heuristic for finding a maximum likelihood estimate of the parameters of the unsupervised model (cluster assignments) and, in most cases, yields an acceptable solution. Lloyd's algorithm is one of the simplest yet one of the most efficient clustering methods. For the sake of the reader's convenience, later mentions of *K-Means* will refer to the cited Lloyd's heuristic.

First, *some* cluster representatives are proposed, often random ones. Then, all the dataset points are assigned a cluster number corresponding to their closest cluster representative. In the next step, cluster representatives are redefined as the mean value of cluster members in each dimension. Since K-Means is a variant of the expectation-maximization algorithm, the assignment step is often referred to as the *expectation step*, and the centroid redefinition step is referred to as the *maximization step*. These expectation and maximization steps are repeated until cluster centers converge to stable values.

K-Means is a general-purpose clustering algorithm with wide possibilities of adjustments to the data characteristics. The adjustments are possible in (but are not limited to) the following areas:

- initialization – custom and deterministic, more and less stable;
- observation similarity – Euclidean, correlation, other custom metrics;
- stop condition – the number of iterations, fixed threshold of centroid displacement, other.

Careful adaptation of the mentioned elements increases the domain relevance of the obtained results compared to the default approach.

The concept of the K-Means algorithm inspired a few other approaches:

- K-Medians [64] – uses median across each dimension instead of mean to compute cluster representatives, minimizes the dataset-wide sum of distances instead of squared distances;

- K-Medoids [68] – the representative is found within the group in a way that minimizes the intra-cluster sum of dissimilarities, often used for distance measures incompatible with classical K-Means algorithm;
- Fuzzy C-Means [46, 18] – cluster membership is expressed as a continuous number from the range $[0, 1]$ and a point may belong to more than one cluster.

### Initialization Method

The simplicity of the K-Means algorithm is also one of its most substantial limitations. Similarly to the way other expectation-maximization algorithms operate, it converges to a local optimum, highly dependent on the algorithm initialization. Inappropriate selection of initial centroids will lead to a local optimum, which does not approximate the true heterogeneity in the data but is a numerical artifact.

**Random Initialization** The simplest methods for initializing K-Means algorithm are Forgy, and Random Partition [57]. The Forgy initialization randomly selects $k$ observations from the dataset and makes them initial cluster centers. The Random Partition initialization assigns a random cluster label to each observation and computes centroids for such a partition. Unfortunately, a comprehensive study of K-Means initialization methods [31] explains that these solutions often perform poorly.

**K-Means++ Initialization** Assuming that it is possible to launch K-Means clustering multiple times, K-Means++ [24] is a very robust stochastic method for initialization. It randomly selects the first initial cluster center from the observations in the dataset. Then, the remaining $k - 1$ initial cluster centers are drawn sequentially at random from the set of observations. The probability of choosing an observation is proportional to the distance between that observation and the already known cluster center closest to it. Although the idea seems worth considering, we did not decide to continue with K-Means++ for Mass Spectrometry Imaging data in the original form due to following reasons:

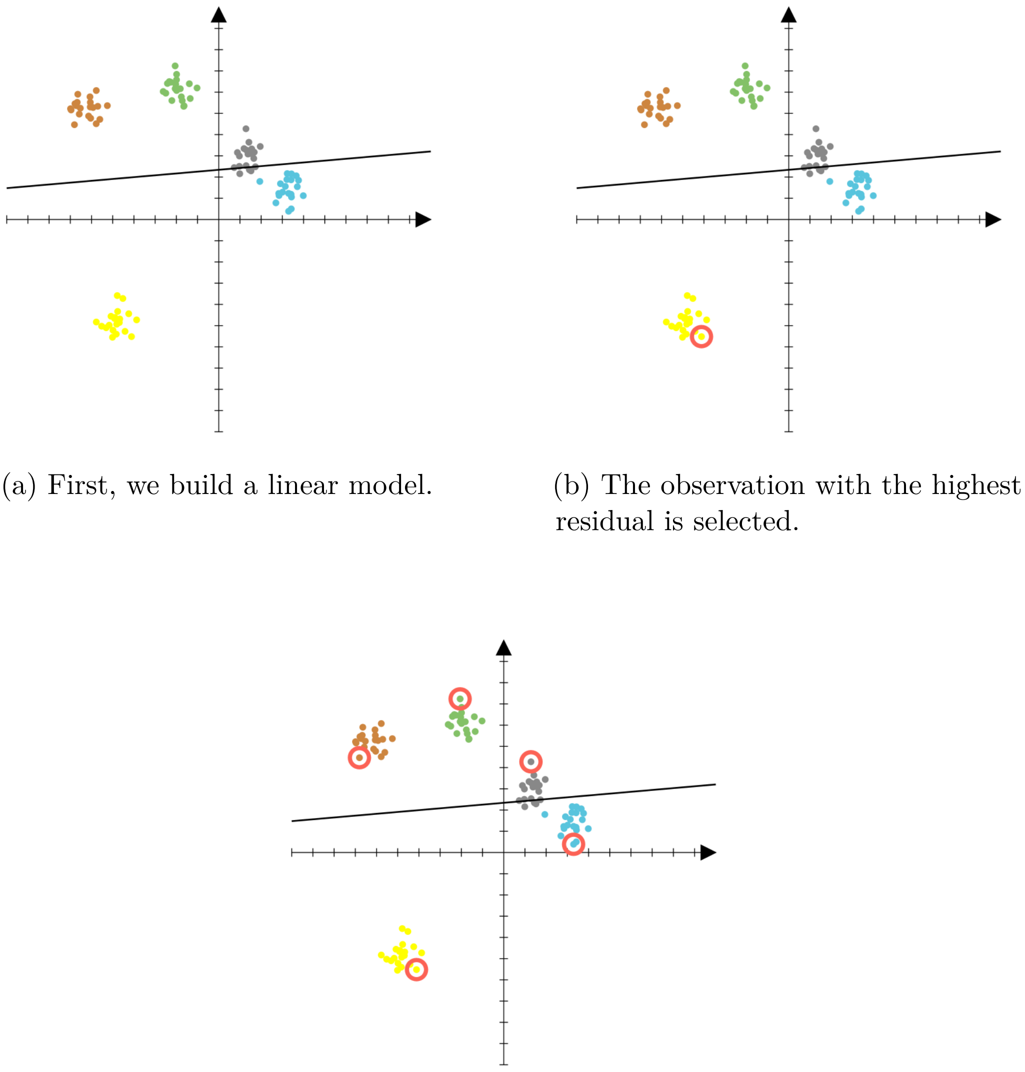
- We aim to capture the molecular heterogeneity, and the K-Means++ only offers *a chance* that the initial cluster centers will be dissimilar from each other.
- It requires multiple launches of the K-Means clustering, which would significantly reduce the scalability of the DiviK framework.

We approached mentioned challenges iteratively with the methods described in the following paragraphs. All the proposed approaches are deterministic to avoid multiple launches.

**Extreme Deterministic Initialization** To ensure that the initial clusters are highly dissimilar, we build a linear model of the dataset and select the observation with the highest residual as the first initial cluster center. Then, similarly to K-Means++ and approaches like [5], we choose the remaining $k-1$ initial cluster centers sequentially. Each time we select the observation, which maximizes the distance between that observation and the already known closest cluster center (see Figure 4.3). The Extreme Deterministic Initialization has two drawbacks:

- low robustness – such a naive algorithm definition is very susceptible to outliers (see Figure 4.5a);
- low scalability – instead of multiple launches of K-Means algorithm, building a linear model increases computational complexity by orders of magnitude.

**Percentile Deterministic Initialization** This initialization replicates the schema of the Extreme Deterministic Initialization but replaces extreme values with a fixed percentile to reduce susceptibility to outliers. We will assume the $95^{th}$ percentile used for this explanation. We build a linear model of the dataset and select the observation with the residual closest to $95^{th}$ percentile



(a) First, we build a linear model.

(b) The observation with the highest residual is selected.

(c) We sequentially select observations furthest from already known centers (green, blue, brown, gray).

Figure 4.3: Demonstration of the Extreme Deterministic Initialization on a synthetic dataset.

as the first initial cluster center. Then, we choose the remaining $k-1$ initial cluster centers sequentially. Each time we select the observation closest to the $95^{th}$ percentile of the set of distances between observations and already known cluster centers closest to them. Other percentiles could be used, but we discourage using percentiles much lower than $90^{th}$ as the dissimilarity of initial centroids may not be sufficient. See Figure 4.4 for detailed diagram flow and Figure 4.5 for demonstration of the algorithm result with synthetic data.

**$k$-d Tree Based Initialization** To address the scalability issues of the previously proposed methods, we applied the $k$-d tree structure. $k$-d tree is a form of a binary tree that contains $k$-dimensional points and partitions the space. We use an unbalanced variation that partitions the space around the mean value in each consecutive dimension, and the dataset points are located in leaves only. The parent nodes of leaves create point clouds which we use instead of original points (observations). The process of constructing such a $k$-d tree is presented in Figure 4.6.

### Distance Measure

K-Means algorithm minimizes the sum of square Euclidean distances between observations and their assigned cluster centers. Therefore it is crucial that although K-Means is flexible concerning distance measure, one cannot use K-Means with *any* distance measure. For such a purpose, one would need to use the K-Medoids algorithm [19] instead. However, there are a few distance measures with proven convergence for the K-Means algorithm. These scenarios still minimize the sum of square Euclidean distances *despite* the other distance measure being used.

**Euclidean Distance** Human perception of distance follows the Euclidean distance. Therefore, it is intuitively understood for two and three dimensions. Since the monotonicity of the Euclidean distance and Euclidean square distance is preserved, the K-Means algorithm requires no modifications to

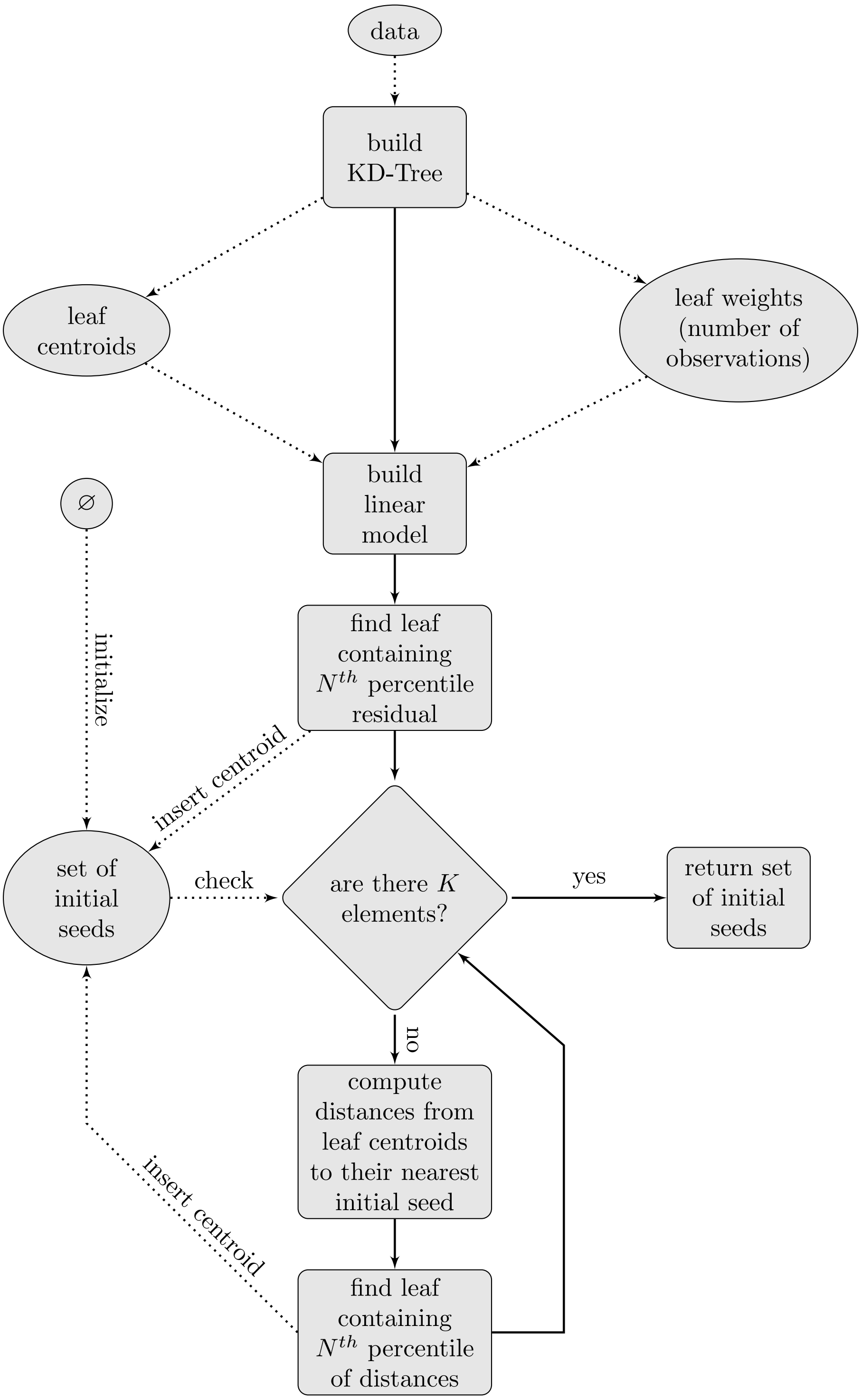
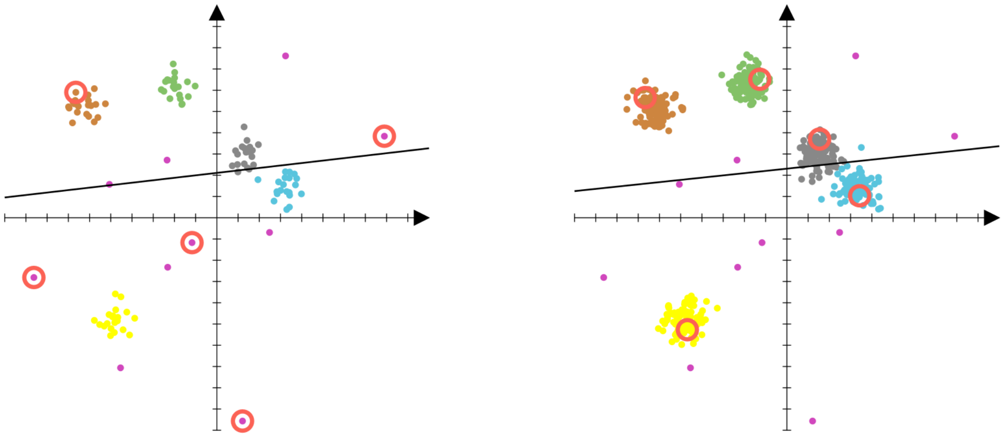

data
build KD-Tree
leaf centroids
leaf weights (number of observations)
build linear model
∅
initialize
find leaf containing $N^{th}$ percentile residual
insert centroid
set of initial seeds
check
are there $K$ elements?
yes
return set of initial seeds
no
compute distances from leaf centroids to their nearest initial seed
insert centroid
find leaf containing $N^{th}$ percentile of distances

Figure 4.4: Flow diagram of percentile deterministic initialization.



(a) Result of extreme initialization for noisy data.

(b) Result of percentile initialization for noisy data. We are using $95^{th}$ percentile for this example.

Figure 4.5: Demonstration of the proposed initialization methods on synthetic datasets.

work with Euclidean distance. Euclidean distance can be computed for any dimensionality with the following formula:

$$d_E(p,q) = \sqrt{\sum_{i=1}^{N}(p_i - q_i)^2} \tag{4.1}$$

where:

$d_E$ Euclidean distance function;

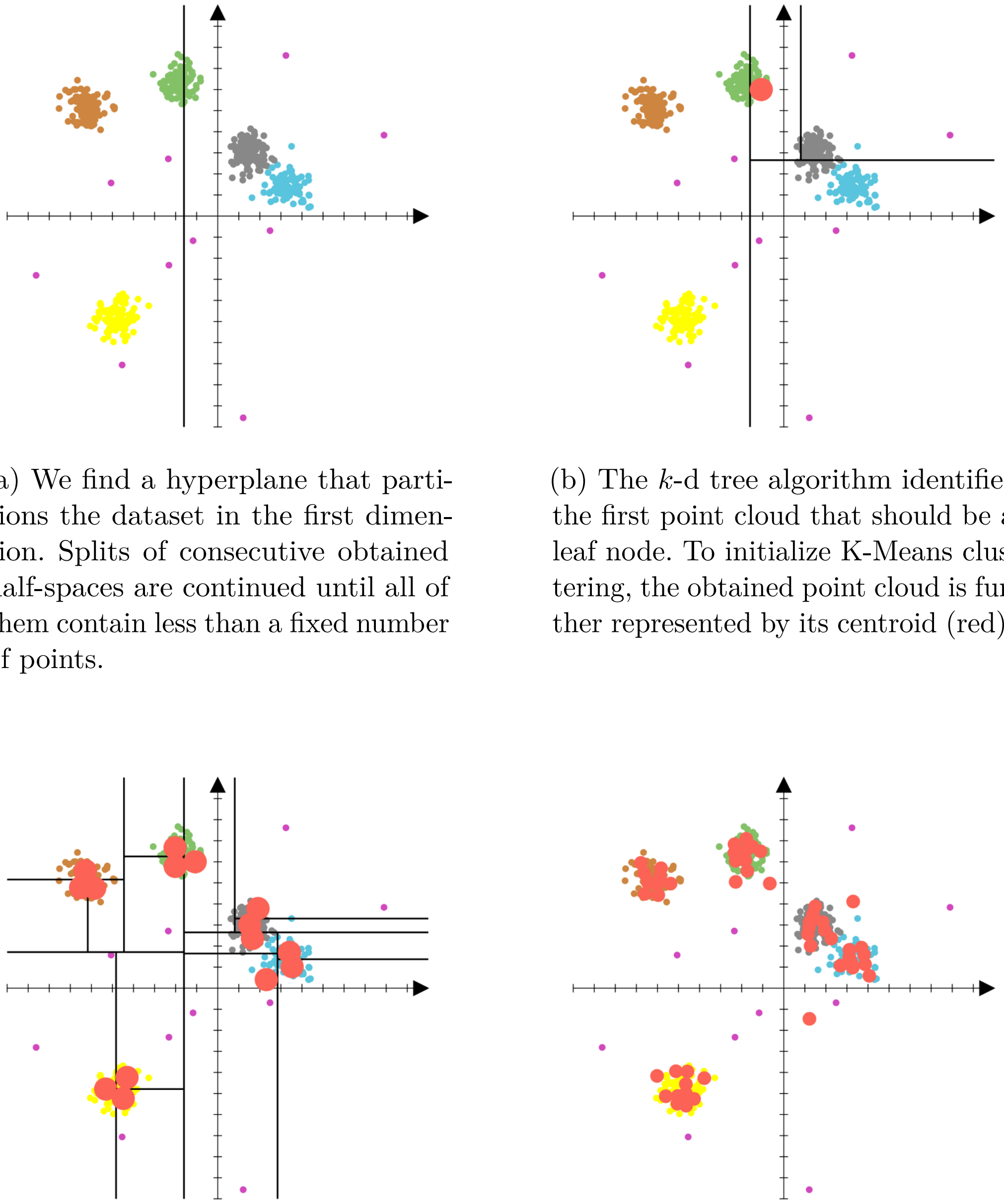
$p, q$ points in $N$-dimensional space;

$N$ number of dimensions;

$p_i, q_i$ coordinates of points $p$ and $q$ in the $i$-th dimension.

Similar notation will be used for other distance measures.

**Cosine Distance** Cosine distance defines the dissimilarity of two points using the angle between vectors connecting the origin of the coordinate system with these points. The exact formula is given in the Equation 4.2:



(a) We find a hyperplane that partitions the dataset in the first dimension. Splits of consecutive obtained half-spaces are continued until all of them contain less than a fixed number of points.

(b) The $k$-d tree algorithm identifies the first point cloud that should be a leaf node. To initialize K-Means clustering, the obtained point cloud is further represented by its centroid (red).

(c) Splits continue until all leaf nodes are identified, and centroids of all point clouds (red) are computed. Here is the complete $k$-d tree built for the synthetic dataset.

(d) One can decrease the size of the point cloud constituting a leaf node. Under such circumstances, centroids of obtained point clouds (red) approximate the original dataset much more precisely.

Figure 4.6: Construction of a $k$-d tree on a synthetic dataset.

$$d_c(p,q) = 1 - \cos\phi = 1 - \frac{\sum\limits_{i=1}^{N} p_i q_i}{\sqrt{\sum\limits_{i=1}^{N} p_i^2 \sum\limits_{i=1}^{N} q_i^2}} \tag{4.2}$$

where $d_c$ is cosine distance function and $\phi$ is the angle between vectors.

A careful reader may notice that the cosine dissimilarity depends entirely on the angle between the vectors, not their length. However, vector length strongly influences the sum of squared Euclidean distances optimized by the K-Means algorithm. Therefore, an additional step is necessary to ensure the convergence of the K-Means algorithm.

To tackle the mentioned issue, one can consider a similar problem with all vectors of unit length. Under such a condition:

$$\begin{aligned}
d_c(p,q) &= 1 - \frac{\sum\limits_{i=1}^{N} p_i q_i}{\sqrt{\sum\limits_{i=1}^{N} p_i^2 \sum\limits_{i=1}^{N} q_i^2}} \\
&= \frac{1}{2}\left(1 + 1 - 2\frac{\sum\limits_{i=1}^{N} p_i q_i}{\sqrt{1 \cdot 1}}\right) \\
&= \frac{1}{2}\left(\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} q_i^2 - 2\sum_{i=1}^{N} p_i q_i\right) \\
&= \frac{1}{2}\sum_{i=1}^{N}(p_i^2 - 2p_i q_i + q_i^2) \\
&= \frac{1}{2}\sum_{i=1}^{N}(p_i - q_i)^2 \\
&= \frac{1}{2} d_E^2(p,q)
\end{aligned}$$

where $d_E^2$ is squared Euclidean distance. The above proves that for unit length vectors, squared Euclidean distance and cosine distance are linearly dependent. Moreover, the K-Means algorithm will be convergent under such an assumption.

Concluding the above reasoning, scaling the original vectors to unit length does not influence pairwise dissimilarities between vectors and is required for convergence of K-Means clustering with cosine distance.

**Pearson Correlation Distance** Pearson correlation distance is conceptually similar to cosine distance. However, it uses the Pearson correlation coefficient instead of the cosine of the angle between vectors. It is given by the following formula:

$$d_P(p,q) = 1 - \frac{\sum_{i=1}^{N}(p_i - \overline{p})(q_i - \overline{q})}{\sqrt{\sum_{i=1}^{N}(p_i - \overline{p})^2 \sum_{i=1}^{N}(q_i - \overline{q})^2}} \tag{4.3}$$

where $d_P$ is Person correlation distance function and $\overline{p}$ is average of the values in $p$.

Similarly to the case of cosine distance, it is important to ensure the convergence of the K-Means algorithm. One can consider a similar problem where $\overline{p} = 0$ for all $p$ in the dataset. Under such assumption, Equation 4.3 reduces to Equation 4.2 and the Pearson correlation distance is equivalent to the cosine distance. As explained previously, that guarantees convergence of the K-Means algorithm.

Concluding the above reasoning, subtracting the mean from each vector does not influence pairwise dissimilarities between vectors and is required for convergence of K-Means clustering with Pearson correlation distance. Then, one needs to follow the normalization process for cosine distance and scale obtained vectors to unit length.

### Stop Condition

There are three most popular stop conditions for the K-Means algorithm:

- number of iterations – algorithm stops after a fixed number of centroid re-computations;
- centroid displacement – algorithm stops if the difference between positions of centroids in consecutive iterations is not greater than a fixed

threshold;

- cluster re-assignment rate – algorithm stops if the rate of cluster label changes is not greater than a fixed threshold.

Calibration of the threshold for centroid displacement highly depends on the feature space and is hard to specify for high-dimensional data. The number of iterations is often used to ensure the end of computations after a fixed time for pessimistic scenarios. In most cases, a few dozens of iterations are sufficient for the K-Means algorithm to converge. Cluster re-assignment rate is loosely connected to centroid displacement but a bit more explainable, regardless of the data dimensionality.

Usually, more than one stop condition is used, especially since it may limit the amount of unnecessary computations. An example of that may be a limit of 100 iterations combined with 0% cluster re-assignment rate, which stops computations right upon convergence.

### Quality Measure

K-Means algorithm is an unsupervised method used for knowledge discovery. Hence the real cluster labels are, in most cases, unknown. In general, quality of the obtained partition may be evaluated via measures of compactness and separation between clusters [56, 22], information criteria [92] or other statistics [119]. However, the cluster definition itself is often arguable. Especially when the clusters overlap and are not clearly separated, which is usually the case for multidimensional data in biological sciences. Therefore, beyond their mathematical definition, such studies are often supported with results of the empirical evaluation, which is critical for real-life performance.

**GAP Statistic** GAP statistic [119] is proposed as a mathematical formalization of statistical folklore, selecting the 'elbow' of the error plot as the right number of clusters in the model. It is designed to apply to any clustering method. However, the authors discuss it mainly on the example of the K-Means algorithm.

At the core of the original formulation of the GAP statistic, there is a sampling from a reference distribution. Given a dataset with $n$ observations, $d$ dimensions, and partition into $k$ clusters, we compute error measures (dispersions) between points and corresponding cluster centers. In parallel, we sample $N$ synthetic datasets with $n$ observations each and uniform distribution in each of $d$ dimensions. The uniform distribution range for each dimension is preserved from the input dataset. Further, each of the $N$ synthetic datasets is partitioned into $k$ clusters to form a reference distribution of the error measure. Finally, we compute the GAP statistic as a dispersion ratio between the input dataset and the reference distribution. Please see Figure 4.7 for the exact flow of the GAP statistic computation and the Equation 4.4 for the definition:

$$Gap_n(k) = E_n^*\{\log(W_k)\} - \log(W_k) \tag{4.4}$$

where:

$Gap_n(k)$ GAP index for $n$ observations partitioned into $k$ clusters;

$W_k$ error measure, squared Euclidean distance between observations and their corresponding cluster centers for the K-Means algorithm case;

$E_n^*$ expectation under a sample of size $n$ from the reference distribution.

To establish the most likely number of clusters in the input dataset, $E_n^*\{\log(W_k)\}$ is computed by averaging the $N$ values of $\log(W_k)$ obtained from clustering the synthetic datasets. As the number of samples $N$ is finite, simulation error is considered:

$$s_k = sd(k)\sqrt{1 + \frac{1}{N}}$$

where:

$s_k$ simulation error;

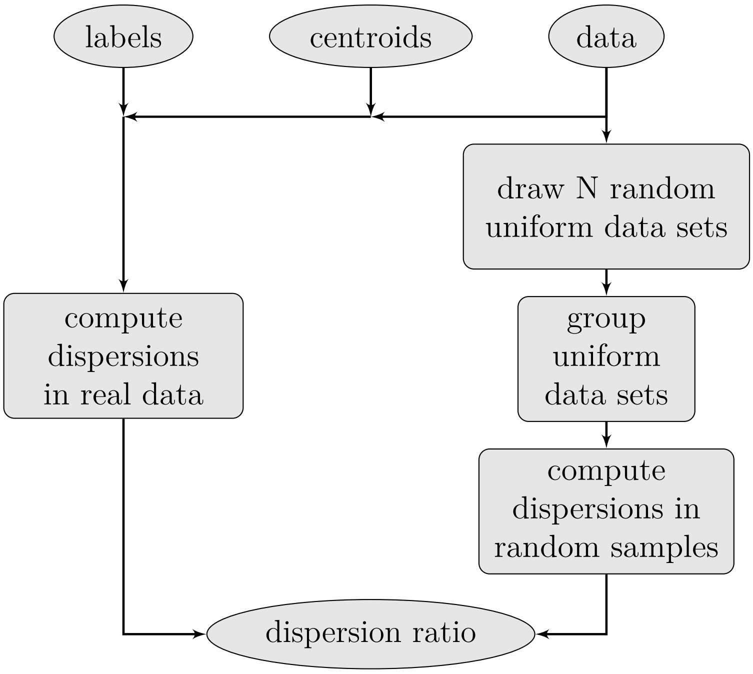

labels
centroids
data
draw N random uniform data sets
compute dispersions in real data
group uniform data sets
compute dispersions in random samples
dispersion ratio

Figure 4.7: Flow diagram explaining how to compute GAP statistic for a dataset partition and a fixed number of clusters.

$sd(k)$ standard deviation of $\log(W_k)$ values obtained from $N$ synthetic datasets.

Values of the GAP statistic and simulation errors are used to find the smallest number of clusters $k$ such that:

$$Gap(k) \geq Gap(k+1) - s_{k+1} \tag{4.5}$$

For the K-Means algorithm case, the authors use the sum of squared Euclidean distances as the dispersion measure. However, another error measure may be more appropriate for other clustering methods.

GAP statistic is one of the very few quality measures which define any value for a single cluster case and allows to decide of whether any split should be conducted. The authors argue that an appropriate statistical estimation of the expected value could be used to achieve similar outcomes for a single cluster case with other quality measures. Although, such a definition often is non-trivial or impossible, depending on the original definition of the clustering quality measure.

The authors discuss a few approaches to defining the reference distribution required for sampling synthetic datasets:

- Uniform distribution – values in each dimension are drawn from the uniform distribution in the same range as the input dataset. As per the authors' empirical study, such an approach appears to be surprisingly effective, despite its apparent simplicity.

- Principal Components Analysis (PCA) – the input dataset is subject to PCA decomposition to make the procedure rotationally invariant for the rotationally invariant clustering methods. Uniform sampling happens in the decomposed space, and the obtained sample is transformed back to the original space of the input dataset. This approach appears to be the most effective in the case of elongated clusters. However, it is slightly less effective for the general case, as it underestimates the number of clusters more often.

- Cluster-level sampling – authors propose two more sampling schemes, based on the above procedures, but conducted separately for each cluster of the input partition. While potentially more precise, unfortunately, this approach has not been evaluated.

Finally, the authors evaluate the risk of recognizing overlapping clusters as a single cluster. Empirical evaluation leads them to the observation that for the overlap proportion of $p$, there is a probability of approximately $p$ for GAP statistic to indicate these clusters are a single cluster.

It must be noted, however, that despite the high relevance of the number of clusters selected with the use of the GAP statistic, there are two major disadvantages of this method:

- Huge computational cost – the number of synthetic datasets $N$ influences the reliability of the obtained results, and values below $N < 10$ are discouraged. This leads to a more than tenfold increased amount of computations required to cluster the data and confirm the number of clusters.

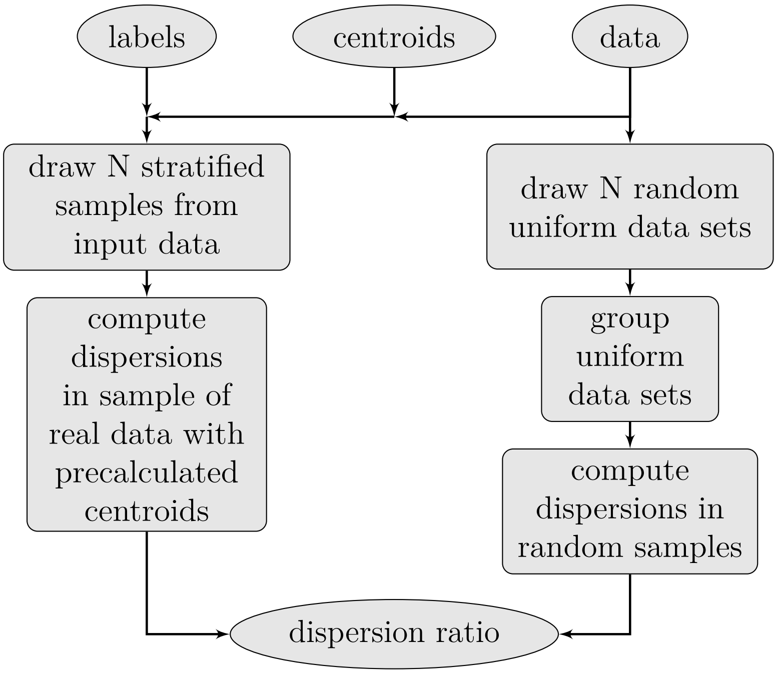

labels
centroids
data
draw N stratified samples from input data
draw N random uniform data sets
compute dispersions in sample of real data with precalculated centroids
group uniform data sets
compute dispersions in random samples
dispersion ratio

Figure 4.8: Flow diagram explaining how to compute sampled GAP statistic.

- Poor scalability in the number of samples – strictly connected with the above limitation. For a high number of observations $n$, the clustering of $N$ synthetic datasets may not be feasible.

**Sampled GAP Statistic** To overcome the computational issues related to calculating the GAP statistic in a classical setting, we propose a sampled GAP statistic. It reduces the computational effort by limiting the number of observations used in the estimations. Instead of sampling only the reference distribution, we sample both the reference distribution and the input data set. We use stratified sampling with respect to cluster assignments obtained with the K-Means algorithm to ensure the representation of each detected substructure regardless of its size. The detailed flow diagram is presented in Figure 4.8.

Since there are two independent sampling processes occurring in parallel, the sampling error $s_k$ must be modified as compared to the original formulation of the method. There are two random variables instead of one: clustering error measure for synthetic datasets $W_k$ and, additionally, clustering error measure for the sampled input data $W_k'$. As the original sampling error $s_k$ was defined based on standard deviation, we propose a similar solution. There are two independent random variables and:

$$Var[X + Y] = Var[X] + Var[Y]$$

$E_n^*\{\log(W_k)\}$ is computed by averaging the $N$ values of $\log(W_k)$ obtained from clustering the synthetic datasets. Similarly for input dataset samples, $E_n^*\{\log(W_k')\}$ is computed by averaging the $N$ values of $\log(W_k')$ obtained from error calculation over the sampled input data. The variance of the estimator defined as the sample mean is given by:

$$Var[\overline{X}] = \frac{\sigma^2}{N}$$

Hence:

$$s_k = \sqrt{\frac{var(k) + var(k')}{N}} \tag{4.6}$$

where:

$s_k$ simulation error;

$var(k)$ variance of $\log(W_k)$ values obtained from $N$ synthetic datasets;

$var(k')$ variance of $\log(W_k')$ values obtained from $N$ samples from original dataset.

Such a sampled formulation of the GAP statistic allows scaling the computations for datasets with millions of observations if subsets of a few thousand examples are drawn to approximate the classical GAP statistic.

**Dunn's Index** The Dunn's index [47] identifies sets of compact and well-separated clusters. It measures the spatial dispersion of the obtained clusters and the distance between these clusters to compare which number of clusters $k$ leads to the optimal separation. The definition is given by the Equation 4.7:

$$D(U) = \min_{\substack{1 \le i \le k \\ 1 \le j \le k \\ i \ne j}} \left\{ \frac{\delta(X_i, X_j)}{\max\limits_{1 \le l \le k} \{\Delta(X_l)\}} \right\} \tag{4.7}$$

where:

$D(U)$ Dunn's index for partition $U$;

$i, j, l$ cluster indices;

$k$ the number of clusters in the partition;

$\delta(X_i, X_j)$ intercluster distance between clusters $X_i$ and $X_j$;

$\Delta(X_l)$ intracluster distance within cluster $X_l$.

Underclustering will decrease the score as the intracluster distance in the denominator will increase for spatially oversized clusters. Overclustering will decrease the score as the intercluster distance in the numerator will decrease for poorly separated clusters. Therefore, selecting the highest Dunn's index value should provide an acceptable trade-off between the under- and overclustering of the dataset.

One of the advantages of Dunn's index is that it is easy for intuitive understanding and interpretation of the results. One could easily imagine Dunn's index threshold, which would lead to an ideal clustering, that should provide an undeniable and clear separation of clusters. This is rarely the case in real-life biological data, especially due to high dimensionality, but Dunn's index may still guide for selecting a non-ideal acceptable solution. Of course, it comes with additional issues, which were mentioned by [56]:

- Dunn's index introduces a considerable amount of time required for computation;

- Dunn's index is sensitive to noise in datasets since it highly influences the intra-cluster distance.

Fortunately, these issues are partly addressed by varying definitions of *intercluster* and *intracluster* distance. [22] gathers six different intercluster distances and three intracluster distances. Each of them comes with different computational complexity and susceptibility to outliers. Please note that the selection of the *intercluster* and *intracluster* distance highly influences the geometrical interpretation of Dunn's index.

Having scalability in mind, we rejected both intercluster and intracluster distance measures, which require computation time quadratic (or higher) in the number of observations. The rejection applies to three out of six intercluster distance measures and two out of three intracluster distance measures. For further considerations we selected *centroid linkage* (Equation 4.8) as the intercluster distance measure and *centroid diameter* (Equation 4.9) as the intracluster distance measure. These measures utilize the information gathered during the clustering process – the location of the cluster centers.

$$\delta_{cl}(S,T) = d(v_s, v_t) \tag{4.8}$$

where:

$\delta_{cl}(S,T)$ centroid linkage between clusters $S$ and $T$ from partition $U$ (intercluster distance);

$v_s$ the centroid of cluster $S$: $v_s = \frac{1}{|S|}\sum\limits_{x \in S} x$;

$v_t$ the centroid of cluster $T$: $v_t = \frac{1}{|T|}\sum\limits_{y \in T} y$;

$|S|, |T|$ the number of elements in clusters $S$ and $T$ respectively;

$d(v_s, v_t)$ the distance between centroids $v_s$ and $v_t$.

$$\Delta_{cd}(S) = 2\left(\frac{\sum\limits_{x \in S} d(x, v_s)}{|S|}\right) \tag{4.9}$$

where:

$\Delta_{cd}(S)$ centroid diameter in the cluster $S$ from partition $U$ (intracluster distance);

$v_s$ the centroid of cluster $S$: $v_s = \frac{1}{|S|} \sum_{x \in S} x$;

$|S|$ the number of elements in cluster $S$;

$d(x, v_s)$ the distance between member $x$ of the cluster $S$ and the centroid $v_s$ of the cluster $S$.

Please note that we do not specify the distance measure between centroids nor between cluster members and centroids. However, we must operate with a normalized dataset to ensure convergence of the K-Means algorithm. Under such circumstances, feasible metrics are linearly related to squared Euclidean distance. Squared Euclidean distance does not satisfy the triangle inequality, and it should be considered when interpreting Dunn's index values with related distances.

**Sampled Dunn's Index** Even though Dunn's index is much less computationally expensive than the GAP statistic, it still scales poorly for massive amounts of observations. Therefore we propose an alternative that works with samples of the input data set.

The original Dunn's index computation can be decomposed into two separate problems: calculation of the smallest intercluster distance and calculation of the biggest intracluster distance. We calculate these measures independently on $N$ samples of $n$ observations. Similarly to the sampled GAP index, we use stratified sampling, as a representation of every cluster is necessary for intercluster and intracluster distances to be defined.

We formulate sampled Dunn's Index as:

$$D_s(U) = \frac{\min\limits_{1 \leq i \leq N} \{\delta_i\} - std(\delta)}{\max\limits_{1 \leq j \leq N} \{\Delta_j\} + std(\Delta)}$$

where:

$D_s(U)$ sampled Dunn's index for partition $U$;

$\delta_i$ minimal intercluster distance for the $i^{th}$ sample subset;

$std(\delta)$ standard deviation of minimal intercluster distances over $N$ sample subsets;

$\Delta_j$ maximal intracluster distance for the $j^{th}$ sample subset;

$std(\Delta)$ standard deviation of maximal intracluster distances over $N$ sample subsets.

Please note that for the *centroid linkage* which we use as a measure of intercluster distance, $std(\delta)$ will always be 0 – we do not sample cluster centers. This measure depends entirely on the parameters of the fitted K-Means model. However, we introduce this component for compatibility with other intercluster distances, which take into account observations directly.

Such a sampled formulation of Dunn's index allows scaling the computations for datasets with millions of observations if subsets of a few thousand examples ($n$) are drawn to approximate the original Dunn's index.

### 4.2.3 Stop Condition

Agglomerative clustering algorithms usually stop when all observations are merged into a single group, and a hierarchy of clusters is known. To obtain a high-level structure of the dataset, it is required to follow the process until the very end. Otherwise, only local similarities will be known. However, Divisive Intelligent K-Means is a deglomerative (divisive) method, and practical scenarios discourage analyses of a few dozens of observations since the result would be dominated by numerical noise. Moreover, it would be a waste of computational resources to produce artificial clustering results without any purpose for further use. Finally, as the feature space changes with each subregion, classical clustering quality measures cannot be applied to assess the entire clustering tree.

There are few approaches in the literature to address the stop condition for divisive clustering algorithms. The simplest ones, like ISODATA, require the user to define a variance threshold that indicates whether a cluster should be further divided. For massively multidimensional data like Mass Spectrometry Imaging, this kind of threshold is hard to calibrate and interpret.

Bisecting variant of the K-Means algorithm requires a predefined number of clusters and splits the cluster with the highest variance. However, in the knowledge discovery applications, the actual number of clusters is unknown upfront and one of the subjects of investigation.

In [108] authors focus on two divisive clustering algorithms:

- the bisecting K-Means algorithm,
- the Principal Direction Divisive Partitioning (PDDP) algorithm [21] – a non-iterative clustering method based on Singular Value Decomposition,

and propose a new method of selecting which cluster should be further divided. The basic criteria – largest cluster size and highest cluster variance – can easily lead to sub-optimal choice for an unbalanced variance of the real clusters and generate numerical artifacts. To address this obstacle, the authors propose an estimation of the cluster's shape, which could be used in conjunction with the aforementioned basic criteria. Computation of the shape measure requires an additional step of splitting the cluster, thus is connected with increased computational cost.

DIANA clustering [91] attempts a reversal of agglomerative hierarchical clustering. Authors select the right number of clusters with the use of the Silhouette index [105] but in the form of post-processing. They do not stop computations until they operate with single-observation clusters. Newer studies in Natural Language Processing for e-learning [89] benefit from this research. Unfortunately, the Silhouette index is not applicable for the assessment of clusters in multiple subspaces at the same time.

Recent research in the multi-criteria decision analysis discipline [63] extends a PROMETHEE clustering algorithm in the form of a divisive clustering method. The *stochastic multi-criteria divisive hierarchical clustering* they propose divides clusters into two new clusters until only one *action* exists or the stopping condition is met. The *action* is an option in the decision process, i.e., how the cluster should be further divided. The case of single *action* is effectively a moment when a cluster reaches the size of two elements, and there is no other way to split it but into singleton clusters. The other stopping condition is based on a *preference degree threshold*. The *preference* *degree* may take values between 0 and 1, where 1 means that some action is absolutely preferred over another action. This relation is not symmetric, and a situation may occur where no action is preferred over a fixed threshold. Under such circumstances, the algorithm would stop. The authors explain that the preference threshold should be set based on domain knowledge. The method is demonstrated to cluster U.S. banks based on financial and non-financial (environmental, societal, governance) criteria of various characteristics and types. The data itself has a hierarchical structure. A similar definition of preference threshold may not yet be possible for Mass Spectrometry Imaging data, as it would require an upfront assessment of thousands of purely numerical features present in the collected data. Moreover, the feature set often changes its meaning between datasets, and these intermediate results of such analysis would not transfer to another study.

Another recent study comes from the domain of resource management of wireless networks [69]. Non-independent and identically distributed data degrade the accuracy of predictions in such domain, and at the same time, privacy concerns are serious. Therefore, even for clustering, a federated learning scenario is employed together with a generative adversarial network. Authors propose a unique divisive clustering algorithm that dynamically adapts the number of clusters that are investigated in the distributed setup, along with two stop conditions:

- all the clusters show a lower variance and mean than the two predefined thresholds;
- the predefined number of iterations is exceeded.

As discussed previously, none of these methods is beneficial for clustering Mass Spectrometry Imaging data.

A more sophisticated method for hierarchical clustering was presented in [7]. IDEA algorithm splits the input dataset into highly-connected chunks based on the nearest neighbor graph. Then, the chunks are clustered hierarchically with multiple linkages simultaneously. Alternative dendrograms are merged in a way that optimizes Dasgupta's cost function [36], and a clustering tree is produced at the output. The obtained clustering tree is cut on the top level, and the original procedure is repeated until a predefined number of subtrees is identified. These subtrees are connected into a final clustering tree with average linkage. The flat clusters are obtained with the repeated cutting of the final tree and elimination of the subtrees below the size threshold as noise-related. The IDEA algorithm gathers different concepts of similarity within a single framework, which provides both the clustering tree and the flat clusters at the output. The computations could be stopped without building the full clustering tree down to a single observation, as it operates with chunks from the nearest neighbor graph throughout most operations. However, it still requires the final number of clusters to be specified upfront.

For the stop condition of the Divisive Intelligent K-Means algorithm, we will focus on two approaches discussed below.

### Subregion Size

The cluster size is the most straightforward stop criterion for the Divisive Intelligent K-Means algorithm. If the cluster size is below a predefined threshold, the algorithm stops processing the corresponding branch of the clustering tree. The stop criterion may consider the nominal cluster size or the initial data set size percentage.

The stop condition based on the percentage of initial dataset size is vital in biological sciences, as it may have understandable interpretation rooted in the domain knowledge. Often a tiny cluster may be just a numerical artifact rather than an actual substructure. Especially when processing an Mass Spectrometry Imaging dataset of low spatial resolution, groups of a few dozens of cells with a unique molecular pattern are unlikely to occur in reality due to averaging effects of the data acquisition aperture.

### GAP Statistic Based

The stop criterion based on GAP statistic follows the idea from [108]. Authors of [108] explain that the shape estimation they propose should work for low dimensionality. However, Mass Spectrometry Imaging data is the opposite – thousands of dimensions are standard. The GAP statistic does not directly estimate the shape of the cluster. However, it may indicate whether there exists any heterogeneity in the cluster. It takes advantage of the fact that the GAP statistic is one of the unique clustering quality measures defined for a single cluster.

Similarly to [108], we first cluster a subset with K-Means algorithm into $k = \{1, 2\}$ clusters. This set of partitions is sufficient to use the Inequality 4.5 and verify whether a single cluster solution is the preferred one in terms of the GAP statistic. We will most likely not obtain the information about the right number of clusters in the subset. However, there is a chance to reject the hypothesis of a single cluster solution and continue the evaluation. In case the GAP statistic provides enough evidence to assume a single cluster solution is valid, the Divisive Intelligent K-Means algorithm stops processing the corresponding branch of the clustering tree. Figure 4.9 presents the detailed flow diagram for this stop condition.

For the sake of scalability, the user may prefer the sampled GAP statistic (see Section 4.2.2) instead of the original GAP statistic, especially for datasets large in terms of the number of observations.

## 4.3 Scalability Considerations for Large Data

A few elements of the Divisive Intelligent K-Means framework contribute to its scalability.

First, filtration-based feature engineering does not require conducting time-consuming transformations like PCA or UMAP. GMM approach is computationally efficient, as it only requires computing the within-cluster variance of each feature. Then, it is followed by an automated data-driven threshold selection, which decomposes the histogram in sub-linear time in the number of samples.

We are using the K-Means algorithm at its core. It is one of the fastest clustering algorithms known. Although in a naive scenario, it may provide unsatisfactory results, in conjunction with local feature space adaptation K-Means demonstrates high relevance of the obtained results.

To ensure the computational complexity of the K-Means algorithm ini-

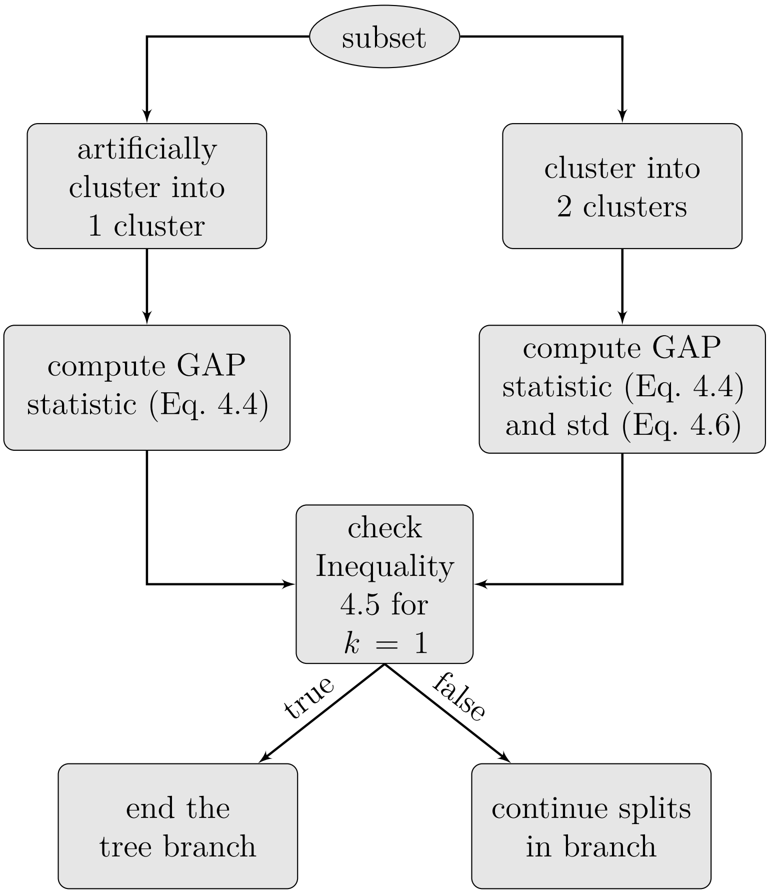

subset
artificially cluster into 1 cluster
cluster into 2 clusters
compute GAP statistic (Eq. 4.4)
compute GAP statistic (Eq. 4.4) and std (Eq. 4.6)
check Inequality 4.5 for $k = 1$
true
false
end the tree branch
continue splits in branch

Figure 4.9: Flow diagram of the GAP statistic based stop condition for the Divisive Intelligent K-Means algorithm.

tialization is linear in the number of samples, we developed a deterministic initialization procedure based on $k$-d tree.

Automated quality assessment during the algorithm operation uses approximation with the sampled GAP statistic and the sampled Dunn's index to speed up computations. Sampled variants of quality measures provide the auto-tuning capabilities without a significant penalty on the scalability.

Finally, we avoid unnecessary computations for homogeneous subregions. The stop condition check with the GAP index provides a criterion to detect such homogeneities and finishes the computations in the related branch of the clustering tree.

Of course, there are more areas in which scalability could be further improved, which are mentioned in Section 6.3.

## 4.4 Experimental Settings

In this work, we compare Divisive Intelligent K-Means to a set of state-of-the-art methods for feature engineering and clustering of Mass Spectrometry Imaging data in all possible pairwise combinations. We select one method of each kind for the comparison. For example, the set of reference clustering methods consists of: K-Means (centroid-based), spectral clustering [41] (density-based), spatial clustering [12] (MSI-dedicated). Similarly, for global feature engineering we use: no global feature engineering (for demonstration purposes), PCA with the knee-based selection of the number of components [107] (linear method), PCA with components selected based on EXIMS-score [133] (MSI-dedicated), UMAP [82] (nonlinear), and neural ion images obtained with a pretrained Xception network (deep learning based) [139]. Along with DiviK, this leads to 20 test combinations of global feature engineering methods and clustering algorithms.

The experiment we conduct has two main areas of investigation:

- Biological relevance of obtained results – the results provided should be at least comparable with other state-of-the-art methods so that we confirm it is possible to discover the details of molecular heterogeneity with the use of the Divisive Intelligent K-Means algorithm.

- Scalability – Divisive Intelligent K-Means should accomplish computations for 3D data in *feasible time* and preserve the relevance of obtained results for massive-volume 3D data.

We consider *feasible time* up to a few days, which poses a severe constraint on the computational complexity with respect to the number of observations in the input dataset. At the same time, such a definition already ensures feasibility in real-life scenarios, as the proper biochemical preparation of dozens of tissue slices for a 3D Mass Spectrometry Imaging experiment is unlikely to span less than a few weeks.

To ensure the reproducibility of the obtained results, all the experiments are conducted in the Polyaxon [85] environment, with source code version control, input data checksum validation, and environment snapshots through Docker.

### 4.4.1 Ground Truth Annotation Translation

The quality assessment for an unsupervised method is particularly complex, as the obtained clusters consist of spectra that the pathologist previously annotated to belong to different tissue regions. Moreover, in a fully unsupervised setup, the number of discovered molecularly heterogeneous regions may differ from the number of regions identified by the pathologist based entirely on the tissue's optical scan. Hence a procedure is required to translate spectrum-level pathologist annotations into cluster-level annotations.

First, we sort clusters descending concerning the percentage of their area covered by a specific Region of Interest (ROI), independently for each ROI (i.e., tumor, healthy epithelium, other tissue). Having clusters organized with such ordering, we approximate the specific ROI with a fixed percentage of clusters and compute the Dice index for the corresponding cluster composition (see Figure 4.10). Based on such a diagram, the composition optimal in terms of the Dice index can be found quickly for each ROI (the red dot on the diagram in Figure 4.10).

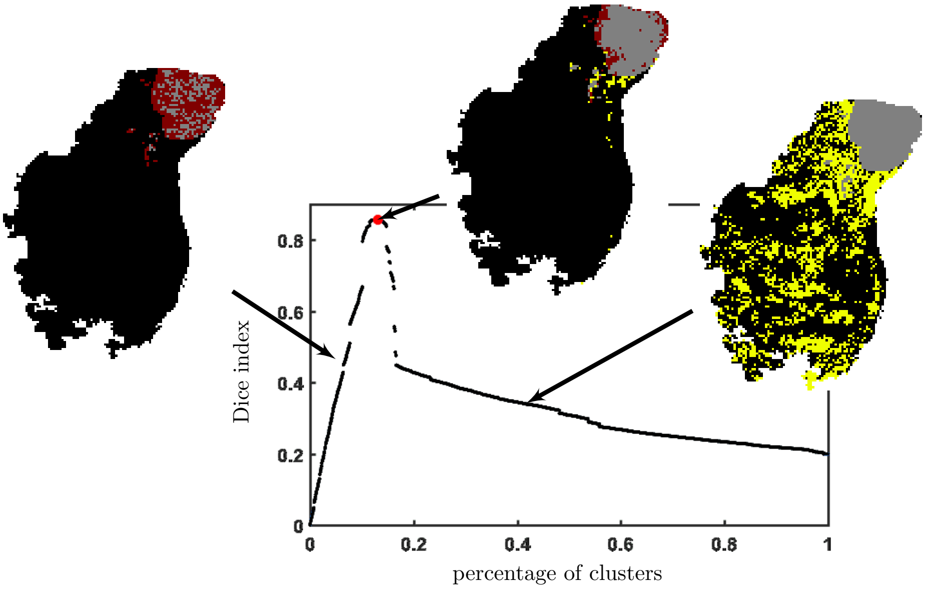


Figure 4.10: ROI approximation process with the binary cluster assignment optimization. We sort clusters descending concerning the percentage of their area covered by a selected ROI. Then, we approximate the ROI with a fixed percentage of clusters and compute the Dice index. The optimal composition can be found in the diagram and is marked with a red dot. Compositions with fewer clusters underestimate ROI (on the left, a region in red is *false negative*). Compositions with more clusters overestimate ROI (on the right, a region in yellow is *false positive*). The Dice index allows us to find the balance between both (in the middle).

However, the above process is a greedy method for binary cluster composition optimization, while in most experimental scenarios, the ground truth has more classes defined. Two ambiguous situations may occur:

- Assignment conflict – as a result of the binary optimization process applied to each ROI separately, a cluster could have been selected as a member of two or more ROIs simultaneously.
- No assignment – as a result of the binary optimization process applied to each ROI separately, a cluster could have been selected as a member of no ROI at all.

For the *crisp clustering* (all observations are assigned exactly one cluster) we are using, both cases require a disambiguation step. The annotation translation process is not supposed to change the definition of the cluster, i.e., clusters cannot become non-crisp. We resolve the issue in a brute-force manner. For all identified ambiguities, we generate a full decision space with different cluster assignment options:

- Assignment conflict – conflicting ROI labels are considered options for a cluster in the generated decision space.
- No assignment – all possible ROI labels present in the ground truth are considered options for a cluster in the generated decision space.

We traverse the decision space, and for each set of possible non-ambiguous cluster assignments, we compute the Rand index with respect to the spectrum-level pathologist annotations. Finally, we select the set of cluster annotations that maximize the Rand index. The cluster interpretation obtained this way is further used for the clustering quality evaluation.

We are aware that the exhaustive search through the full decision space may not be a scalable process; however, it should not be an issue for real-life applications due to the following reasons:

- The ground truth annotation translation is required only for quality assessment of an algorithm if one wants to compute precise values of quality metrics. It is not used as a part of the clustering algorithm. This translation process will not be used for unsupervised knowledge discovery and absent ground truth labels. We describe it for complete clarity and transparency on the origin of the quality measures presented further in this work.
- The first step of binary optimization with the use of the Dice index already provides very efficient pruning of the decision space. We observed the necessity to evaluate up to 30 disambiguation scenarios for the partitions obtained in this work, which could be accomplished in up to a minute.

### 4.4.2 Evaluation Criteria

#### Biological Relevance

To assess the biological relevance of the obtained results, we must use a dataset with ground truth labels provided by an experienced pathologist. The OSCC dataset described in Section 2.2 fits that purpose.

There are different kinds of *biological relevance* considered in the literature discussed in Chapter 3. We try to capture these ideas with the following numerical metrics:

- Rand index – as a way to measure global multi-ROI reconstruction capabilities;
- Dice index – as a way to measure tumor reconstruction capabilities of a clustering algorithm;
- EXIMS score [133] – as a way to measure the spatial consistency of the clusters and their biological likelihood.

The EXIMS score is a measure with unbounded value, valid for comparative analyzes, but the magnitude alone is hard to interpret. To provide a point of reference, we scale and clip values so that the highest *relative EXIMS score* for the non-singleton partition can be 1.

Note that such a definition of relative EXIMS score enables a valid comparison of a few experiments conducted as a part of one study. However, any comparison between studies would be unjustified. It is not impossible but requires recalculation of relative EXIMS scores between the studies to use consistent scales.

Additionally, we define two measures of *overall quality* as supportive quantities to indicate the trade-off between the above scores. Each pair of feature engineering and clustering methods evaluated with the OSCC data in this work can be represented as a point in a three-dimensional space. The coordinates of such a point are defined by Dice index, Rand index, and relative EXIMS score.

- The distance between the point representing the combination of methods and the origin of the coordinate system is the *overall quality* $d(0,0,0)$.
- The distance between the point representing the combination of methods and the theoretical maximum of each score $(1,1,1)$ is the *overall quality* $d(1,1,1)$.

#### Scalability

Two aspects are essential to claim that a clustering method is scalable: the possibility to complete computations in a feasible time and no significant degradation in the relevance of the obtained results for large-scale data.

Since the 3D benchmark datasets featured in the literature do not introduce publicly available labels of functional regions of the tissue, biological relevance for the 3D data cannot be numerically assessed as described in Section 4.4.2. Therefore, in this area, we will follow other researchers and visually compare the discovered tissue structure with other studies.

The aspect of *possibility to complete computations in feasible time* holds requirements not only towards the time complexity of the method but also space complexity. This observation can be used as a preliminary indication of whether a method would fail to complete the computations within the imprecise limit of *few days*, as defined at the beginning of Section 4.4. Practically, if an implementation of an algorithm attempts to allocate petabytes of memory (or more), its memory and computational complexity are beyond scalability limits.

### 4.4.3 Hyperparameter Settings

#### Oral Squamous Cell Carcinoma Dataset

We set the threshold for the minimal number of features to be preserved during GMM-based filtering in DiviK to 1% (which corresponds to at least 37 locally most relevant features). K-Means algorithm in DiviK sweeps from 1 to 10 clusters on each level of the segmentation hierarchy tree. $k$-d tree based initialization approximates the OSCC dataset with leaves of size not bigger than 1% of the initial dataset size (which corresponds to at most 198 spectra averaged on the top level of the clustering tree). The algorithm starts from the leaf containing $99^{th}$ percentile of the distance. We use correlation distance, as it is confirmed to provide meaningful results for Mass Spectrometry Imaging data [41, 114]. To compute the sampled Dunn's index and the sampled GAP statistic, we sample 10 times 1,000 spectra each to keep the computational complexity of quality estimation bounded. The stop condition related to subregion size in DiviK ends the computations when 200 spectra or less are present in a considered subregion.

We launch standalone K-Means clustering sweeping up to 50 clusters to provide an additional margin for capturing molecular heterogeneity in the data. Criteria for computing the sampled GAP statistic are identical to DiviK ones. Spatial clustering also sweeps up to 50 clusters and is launched with the bilateral filter of radius 7. Spectral clustering is used with cosine metric during the embedding, precisely as described in the original publication [41]. The embedding generates the number of components equal to 1% of the initial number of features to ensure information capacity comparable with filtering in DiviK.

UMAP embedding uses 30 neighbors during graph construction and the correlation distance. We increase the number of epochs to 500 for increased precision (as compared to 200, which is the default) and a negative sample rate of 70. At the output, we obtain three components. Note that in the clustering scenarios operating with UMAP-embedded data, we switch to the Euclidean distance. The correlation distance is already represented in the embedding, and using it for clustering would be unjustified.

We use Xception neural network with a patch size of 71x71 pixels due to the low spatial resolution in the OSCC dataset. Moreover, it requires an upsample rate of 4 to obtain the patch edge length of $1.775mm$, which lies in the range recommended by the authors of the method [139]. The neural ion images are generated from the U-D pipeline proposed by the authors.

### Mouse Kidney 3D Dataset

Due to the massive volume of the mouse kidney dataset in the number of samples, minor modifications are applied as compared to the OSCC dataset.

DiviK is set to preserve at least 0.5% of initial features, corresponding to 38 locally most relevant features. $k$-d tree leaf size upper bound is set to 0.1% of the subregion size (which corresponds to at most 1362 spectra averaged on the top level of the clustering tree). The algorithm starts from the leaf containing $95^{th}$ percentile of the distance. To compute the sampled Dunn's index and the sampled GAP statistic, we sample 10 times 5,000 spectra each to keep the computational complexity of quality estimation bounded. As the tissue labels are missing in the dataset and the detailed tissue heterogeneity cannot be confirmed, we stop computations for a subregion that contains 50,000 spectra or fewer out of the initial 1,362,830 (approximately 3.6% of the total dataset size).

The neural ion images method for feature engineering [139] lacks generalization for 3D data at the moment and thus was omitted in the comparison.

The hyperparameters of other algorithms included in the comparison remain unchanged.

# 4.5 Results and Discussion

## 4.5.1 Oral Squamous Cell Carcinoma Dataset

Visualization of clustering results for all combinations of feature engineering and clustering methods is presented in Figure 4.11. As one can see, algorithms exhibit varying capabilities to discover biologically relevant tissue regions. Few combinations completely missed clusters related to the tumor, i.e., Knee PCA combined with K-Means, UMAP combined with spatial clustering, and neural ion images (Xception) combined with spectral clustering. Due to the low medical relevancy of obtained results for these three cases, we bind their relative EXIMS score at 1.00 for further consideration.

Such a result suggests an urgent need for careful design and throughout validation of different combinations of methods since both UMAP and spatial

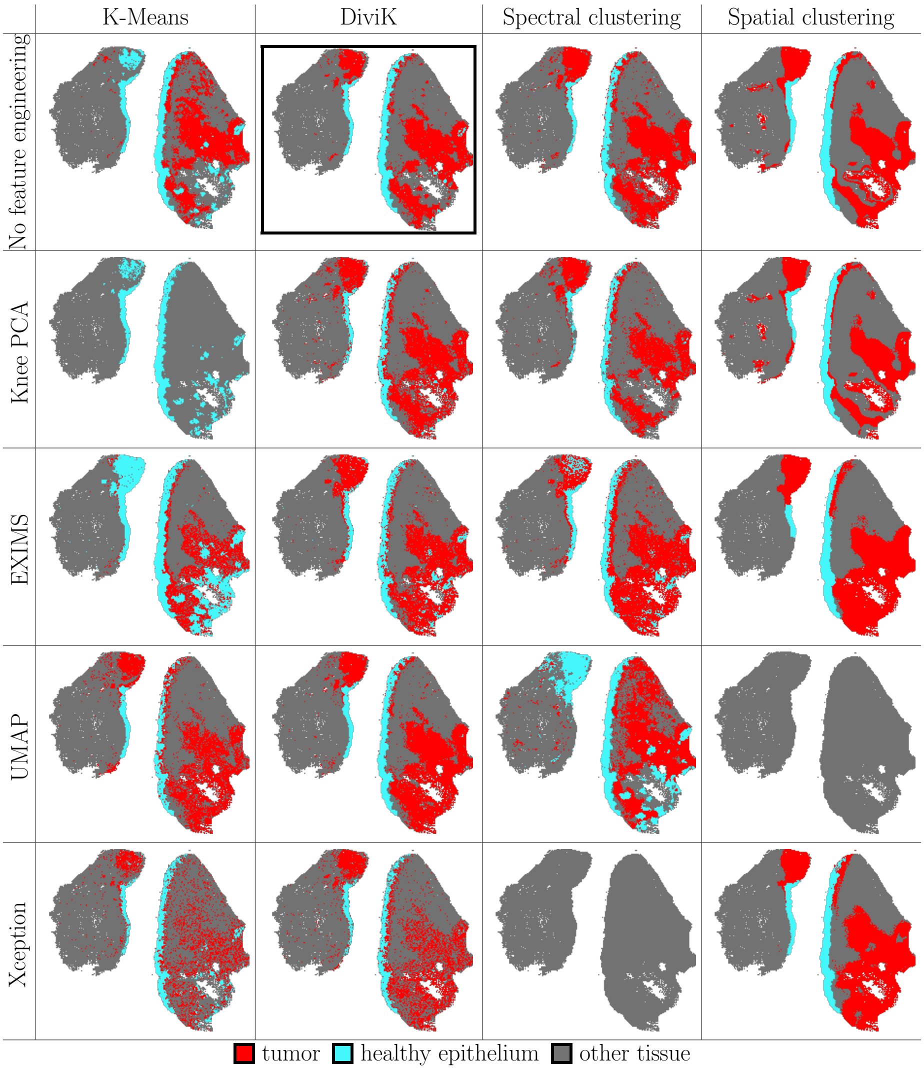


Figure 4.11: The partitions obtained with all combinations of selected feature engineering and clustering methods for OSCC data. Feature engineering methods in rows, clustering methods in columns. Presented regions correspond to ROIs defined by pathologist: red – tumor, cyan – healthy epithelium, gray – other tissue (compare Figure 2.6). The cluster annotations were obtained with the label translation process described in Section 4.4.1. Additional frame is applied around the partition with the top *overall quality* $d(0, 0, 0)$.

clustering tend to exhibit the vast potential to capture all necessary details – equally in this study and domain literature discussed in Chapter 3. One likely explanation for such an invalid result could be the conceptual incompatibility of the two algorithms. UMAP reduces feature space to just three dimensions through the embedding. While useful for data visualization and simpler clustering algorithms, it discards most of the nuances in the dataset. At the same time, these nuances could introduce additional sparsity to the data and could be considered *edges* of the ion images by the bilateral filter, which is an integral part of the spatial clustering method.

A similar observation can be made for the combination of neural ion images (Xception) and spectral clustering. Neural ion images may change the original distribution of the MSI data, which characteristic was one of the assumptions for the spectral clustering as parameterized and described in [41].

After visual inspection, much better effectiveness can be claimed for EXIMS-based feature engineering combined with spatial clustering. This effectiveness can be confirmed with the exact values of quality measures, which are gathered in Table 4.1 for all combinations of methods. For convenience, we visualize the quality indices in Figure 4.12. EXIMS-based feature engineering combined with spatial clustering yields the top Dice index and the top global ROI composition as expressed with the Rand index. The strong synergy between both approaches probably causes such a great result. EXIMS-based feature engineering provides a set of 8 biologically plausible features, significantly limiting the amount of noise at the same time. Spatial clustering has enough sparsity in the data for finding the edges between molecularly heterogeneous regions, but at the same time, the edges are less noisy than for other scenarios.

The second-highest Dice index and the third-highest Rand index occur for 43 neural ion images obtained with the use of the Xception network, clustered with spatial clustering. This result would support the above hypothesis about the synergy between methods, as both pipelines with EXIMS-based feature engineering and the neural ion images are conceptually very similar. However, these are not the only evaluation criteria for medical experts interpreting the results.

Table 4.2 gathers the ranks for the selected quality measures and presents row-wise rank sums. As expected upon visual inspection, spatial clustering

Table 4.1: Clustering quality measures computed for OSCC data. Zero Dice index occurs for partitions with no tumor region separated (compare Figure 4.11).

| clustering algorithm | global feature engineering method | adjusted Rand index | Dice index | relative EXIMS score |
|---|---|---|---|---|
| Spectral | UMAP | 0.2792 | 0.4844 | 0.5891 |
| Spatial | UMAP | 0.0000 | 0.0000 | 1.0000 |
| Spectral | Xception | 0.0000 | 0.0000 | 1.0000 |
| K-Means | Knee PCA | 0.2723 | 0.0000 | 1.0000 |
| K-Means | Xception | 0.3098 | 0.4577 | 0.9197 |
| K-Means | EXIMS PCA | 0.4827 | 0.5129 | 0.8323 |
| Spectral | EXIMS PCA | 0.5447 | 0.7418 | 0.6449 |
| K-Means | none | 0.3364 | 0.5043 | 0.9712 |
| K-Means | UMAP | 0.5231 | 0.7238 | 0.7225 |
| Spatial | Knee PCA | 0.4985 | 0.7065 | 0.7639 |
| DiviK | EXIMS PCA | 0.6082 | 0.7765 | 0.6383 |
| Spectral | none | 0.5906 | 0.7966 | 0.6520 |
| DiviK | Knee PCA | 0.5567 | 0.7540 | 0.7289 |
| DiviK | Xception | 0.4203 | 0.6429 | 0.9395 |
| Spatial | none | 0.5617 | 0.7720 | 0.7587 |
| DiviK | UMAP | 0.6534 | 0.8369 | 0.6568 |
| Spatial | Xception | 0.6517 | 0.8465 | 0.6851 |
| Spectral | Knee PCA | 0.4594 | 0.6897 | 0.9891 |
| Spatial | EXIMS PCA | 0.7035 | 0.8672 | 0.6977 |
| DiviK | none | 0.5433 | 0.7372 | 1.0000 |

with EXIMS and spatial clustering with Xception network yield the two best results out of all combinations of methods. DiviK without feature engineering provides the third best result due to the most biologically plausible structure expressed with EXIMS score and relatively precise cluster delineation. One can observe the sum of ranks ranging from 16 to 53. DiviK without feature engineering was assigned a rank of 20.5, around 12% of the observed range.

Table 4.2 may lead to another observation that algorithms tend to perform well either with Rand and Dice index simultaneously or EXIMS score. Spatial clustering with PCA seems an exception to this rule (the rank of 11 for Rand

Table 4.2: Quality measures ranked for OSCC data. Lower rank is better.

| clustering algorithm | global feature engineering method | adjusted Rand index rank | Dice index rank | relative EXIMS score rank | sum of ranks |
|---|---|---|---|---|---|
| Spectral | UMAP | 17 | 16 | 20 | 53 |
| Spatial | UMAP | 19.5 | 19 | 2.5 | 41 |
| Spectral | Xception | 19.5 | 19 | 2.5 | 41 |
| K-Means | Knee PCA | 18 | 19 | 2.5 | 39.5 |
| K-Means | Xception | 16 | 17 | 8 | 41 |
| K-Means | EXIMS PCA | 12 | 14 | 9 | 35 |
| Spectral | EXIMS PCA | 8 | 8 | 18 | 34 |
| K-Means | none | 15 | 15 | 6 | 36 |
| K-Means | UMAP | 10 | 10 | 13 | 33 |
| Spatial | Knee PCA | 11 | 11 | 10 | 32 |
| DiviK | EXIMS PCA | 4 | 5 | 19 | 28 |
| Spectral | none | 5 | 4 | 17 | 26 |
| DiviK | Knee PCA | 7 | 7 | 12 | 26 |
| DiviK | Xception | 14 | 13 | 7 | 34 |
| Spatial | none | 6 | 6 | 11 | 23 |
| DiviK | UMAP | 2 | 3 | 16 | 21 |
| Spatial | Xception | 3 | 2 | 15 | 20 |
| Spectral | Knee PCA | 13 | 12 | 5 | 30 |
| Spatial | EXIMS PCA | 1 | 1 | 14 | 16 |
| DiviK | none | 9 | 9 | 2.5 | 20.5 |

and Dice indices, the rank of 10 for relative EXIMS score), with the sum of ranks 32 (43% of the ranks range).

In terms of *overall quality* $d(0, 0, 0)$ (see Table 4.3), the Divisive Intelligent K-Means method combined with no additional feature engineering is the top one. According to the purpose of the *overall quality* $d(0, 0, 0)$ measure, this means that DiviK provides a reasonable trade-off between absolute ROI reconstruction capabilities from the molecular information and the obtained spatial composition of the clusters. The obtained tumor region is investigated in correlated ion images (see Figure 4.13), and the corresponding ions are identified. These ions may signalize the energy production process

Table 4.3: Overall quality describing trade-off between quality measures from Table 4.1. Preferred values are marked with gray background.

| clustering algorithm | global feature engineering method | overall quality $d(0,0,0)$ | overall quality $d(1,1,1)$ |
|---|---|---|---|
| Spectral | UMAP | 0.8122 | 0.9768 |
| Spatial | UMAP | 1.0000 | 1.4142 |
| Spectral | Xception | 1.0000 | 1.4142 |
| K-Means | Knee PCA | 1.0364 | 1.2368 |
| K-Means | Xception | 1.0730 | 0.8814 |
| K-Means | EXIMS PCA | 1.0903 | 0.7300 |
| Spectral | EXIMS PCA | 1.1237 | 0.6325 |
| K-Means | none | 1.1449 | 0.8288 |
| K-Means | UMAP | 1.1487 | 0.6170 |
| Spatial | Knee PCA | 1.1537 | 0.6272 |
| DiviK | EXIMS PCA | 1.1749 | 0.5782 |
| Spectral | none | 1.1868 | 0.5745 |
| DiviK | Knee PCA | 1.1873 | 0.5749 |
| DiviK | Xception | 1.2136 | 0.6835 |
| Spatial | none | 1.2195 | 0.5498 |
| DiviK | UMAP | 1.2485 | 0.5143 |
| Spatial | Xception | 1.2691 | 0.4940 |
| Spectral | Knee PCA | 1.2904 | 0.6235 |
| Spatial | EXIMS PCA | 1.3167 | 0.4438 |
| DiviK | none | 1.3560 | 0.5269 |

characteristic for oncologic issues.

At the same time, the *overall quality* $d(1,1,1)$ indicates Spatial clustering with EXIMS PCA-based feature engineering as the top result. Spatial clustering over Xception-generated features and DiviK clustering with UMAP feature engineering yield the next two best scores. *Overall quality* $d(1,1,1)$ ranges from 0.4438 to 1.4142, and DiviK with UMAP feature engineering was around 7% of the observed quality measure range.

Note that we do not aim to find the combination of methods top in terms of any specific quality measure. The study is to verify if DiviK can provide a

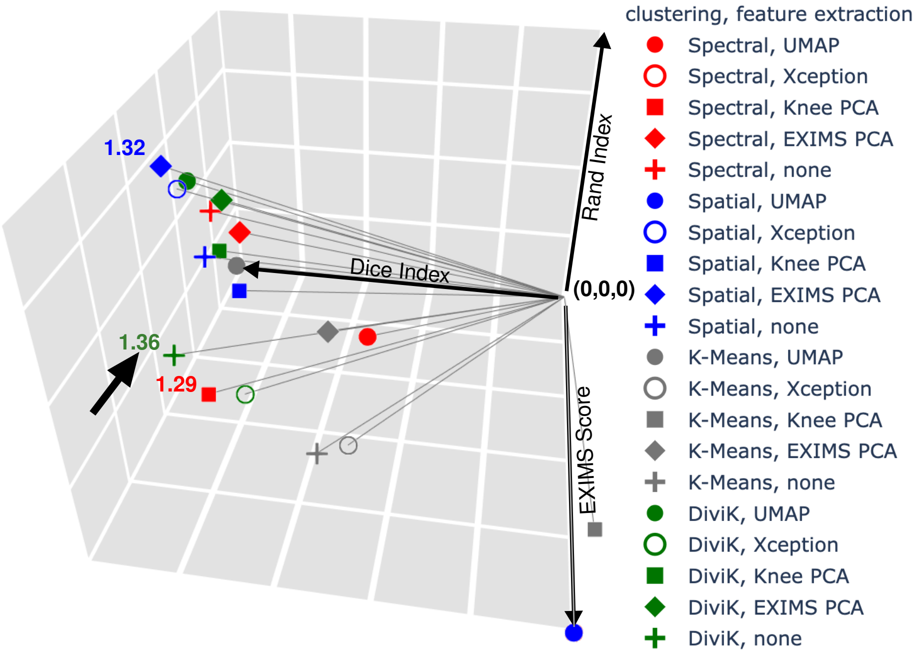


Figure 4.12: Graphical representation of clustering quality measures computed for OSCC data presented in 3D space with coordinates given by Dice index, Rand index and relative EXIMS score. All points are connected to the origin of the coordinate system, and the length of this segment is the *overall quality* $d(0, 0, 0)$ for a given combination of methods. Arrow points the top combination in terms of *overall quality* $d(0, 0, 0)$.

similar level of insights compared to state-of-the-art methods and scales well for massive data volumes simultaneously.

The Divisive Intelligent K-Means method seems to yield the most consistent results regardless of the feature engineering method applied. This statement is supported by visual comparison of clusters in Figure 4.11 with ground truth from Figure 2.6. One could explain it with the fact that each of the other algorithms missed the tumor once in our investigations. In Table 4.4 we summarize the quality measures of each clustering algorithm in our evaluation and support that also claim numerically – DiviK yields the lowest standard deviation in terms of Dice index and Rand index.

The results most visually similar to DiviK were obtained using spectral clustering with no feature engineering or knee PCA. On the other hand,

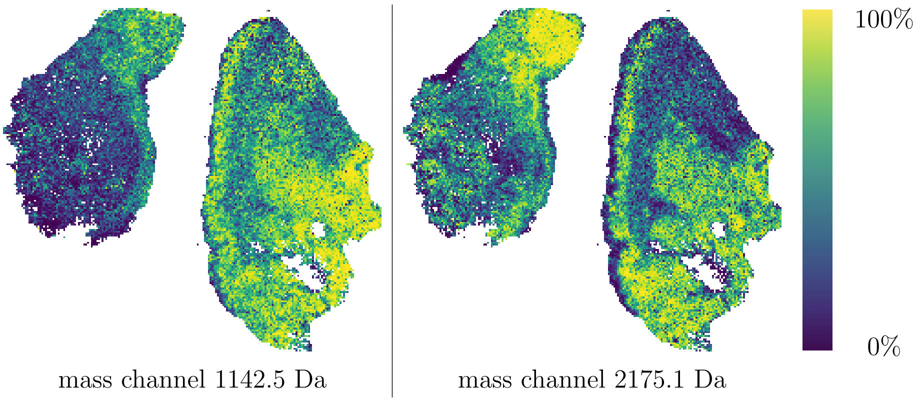


Figure 4.13: Sample upregulated peptides correlated with tumor region as discovered via DiviK. In the figure, one can see peptides with 1142.5 m/z and 2175.1 m/z. They are putatively fragments of pyruvate kinase, an enzyme involved in the Warburg effect.

Table 4.4: Clustering quality measures for OSCC data averaged by clustering algorithm. Standard deviation in brackets.

| clustering algorithm | adjusted Rand index | Dice index | relative EXIMS score | overall quality $d(0,0,0)$ | overall quality $d(1,1,1)$ |
|---|---|---|---|---|---|
| Spectral | 0.375<br>(0.241) | 0.543<br>(0.325) | 0.775<br>(0.202) | 1.083<br>(0.184) | 0.844<br>(0.357) |
| K-Means | 0.385<br>(0.111) | 0.440<br>(0.266) | **0.889**<br>**(0.113)** | 1.099<br>**(0.048)** | 0.859<br>(0.234) |
| Spatial | 0.483<br>(0.281) | 0.638<br>(0.363) | 0.781<br>(0.127) | 1.192<br>(0.123) | 0.706<br>(0.402) |
| DiviK | **0.556**<br>**(0.088)** | **0.750**<br>**(0.071)** | 0.793<br>(0.167) | **1.236**<br>(0.073) | **0.576**<br>**(0.067)** |

Xception-based feature engineering reduces the consistency of regions obtained with K-Means and DiviK. Two hypotheses could lead to an explanation:

- Noise amplification – Xception neural network is used to group images representing structures similar visually, but contrary to EXIMS, does not validate whether the content provides any biologically plausible structure. Such an approach, connected with the deduplication of original features, may cause relative amplification of noise compared to meaningful information.

- Insufficient ion image preprocessing – the authors propose ion image winsorization and scaling. However, despite such efforts, the contrast of the patches may be too low to obtain relevant embedding. The idea behind neural ion images is to mimic a human reviewer manually grouping the original ion maps. Applying the classical ion image normalization schema [98] would probably be beneficial for the end results. A comparison of both preprocessing methods on sample ion image is present in Figure 4.14.

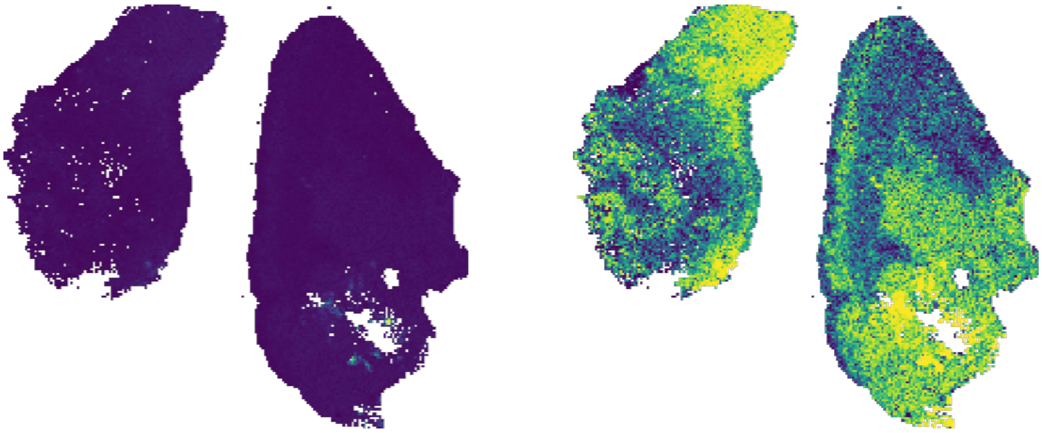


Figure 4.14: Comparison of ion map contrast enhancement methods. In the figure one can see the peptide with 2175.1 m/z (like in Figure 4.13). In the left panel, we apply winsorization and scaling as authors of [139], while in the right panel, we apply histogram equalization as authors of [98].

In Table 4.5 we gather clustering quality measures averaged by the feature engineering method. Scenarios with no feature engineering at all demonstrate the highest *overall quality* $d(0, 0, 0)$. This result may indicate that most of the tested clustering methods are sufficiently suited for processing high-dimensional Mass Spectrometry Imaging data to discard feature engineering method unless explicitly required, e.g., due to the processing time. *Overall quality* $d(1, 1, 1)$ leads to similar conclusion for no feature engineering. However, it also shows the increased performance of the EXIMS-based feature engineering.

Table 4.5: Clustering quality measures for OSCC data averaged by feature engineering algorithm. Standard deviation in brackets.

| global feature engineering method | adjusted Rand index | Dice index | relative EXIMS score | overall quality $d(0,0,0)$ | overall quality $d(1,1,1)$ |
|---|---|---|---|---|---|
| UMAP | 0.364<br>(0.288) | 0.511<br>(0.371) | 0.742<br>(0.180) | 1.052<br>(0.190) | 0.881<br>(0.407) |
| Xception | 0.346<br>(0.271) | 0.487<br>(0.361) | **0.886**<br>**(0.138)** | 1.139<br>(0.124) | 0.868<br>(0.397) |
| Knee PCA | 0.447<br>(0.123) | 0.538<br>(0.359) | 0.871<br>(0.144) | 1.167<br>(0.105) | 0.766<br>(0.315) |
| EXIMS PCA | **0.585**<br>**(0.094)** | **0.725**<br>**(0.151)** | 0.703<br>(0.090) | 1.176<br>(0.100) | **0.596**<br>**(0.119)** |
| none | 0.508<br>(0.116) | 0.703<br>(0.134) | 0.846<br>(0.168) | **1.227**<br>**(0.091)** | 0.620<br>(0.141) |

Further investigation of the averaged results for feature engineering methods is much less conclusive than for clustering methods. A simple rule for outstanding stability or effectiveness cannot be easily formulated. Note that EXIMS PCA and no feature engineering provide increased biological relevance of obtained results but at the same time provide a sub-optimal cluster consistency at the output.

We conduct the effect size analysis to assess the impact of the feature engineering method and clustering method on the quality measures qualitatively. Using the results from Table 4.1, we perform the Kruskal-Wallis test (a non-parametric equivalent for analysis of variance). Results from the test are used to calculate the partial $\eta^2$ effect size. Partial $\eta^2$ measures the proportion of the variance explained by a given variable to the total variance in the model, remaining after considering the variance explained by all other variables. The obtained values for partial $\eta^2$ are presented in the Table 4.6. In statistics, partial $\eta^2$ of 0.06 is assumed at least *medium effect size*, and partial $\eta^2$ of 0.14 or more is assumed *large effect size*. Except for the impact of the clustering method on relative EXIMS score (medium effect size), all other obtained effect sizes are large.

Table 4.6: Effect size analysis for assessing the impact of feature engineering and clustering methods on quality measure differences observed in Table 4.1. We compute the partial $\eta^2$ in the table with the results of the Kruskal-Wallis test. Below, we present Kendall's W concordance index for the same difference rankings.

| Quality measure | Feature engineering | Clustering |
|---|---|---|
| Partial $\eta^2$ | | |
| Rand index | 0.203 | 0.258 |
| Dice index | 0.161 | 0.293 |
| EXIMS | 0.262 | 0.095 |
| Overall quality $d(0,0,0)$ | 0.141 | 0.345 |
| Overall quality $d(1,1,1)$ | 0.122 | 0.309 |
| Kendall's W Concordance Index | | |
| Rand index | 0.328 | 0.325 |
| Dice index | 0.328 | 0.325 |
| EXIMS | 0.136 | 0.375 |
| Overall quality $d(0,0,0)$ | 0.424 | 0.138 |
| Overall quality $d(1,1,1)$ | 0.472 | 0.138 |

Additionally, we support the effect size analysis with Kendall's W concordance index, which measures agreement among different raters as a value in the range of $[0,1]$. It is a non-parametric equivalent of the correlation coefficient. The higher the value, the more agreement between raters; intermediate values indicate greater or lesser consensus. We can observe the most consensus on the overall quality, both $d(0,0,0)$ and $d(1,1,1)$, when using feature engineering methods as raters. The next highest consensus is on the cluster spatial consistency expressed as relative EXIMS score when using clustering methods as raters. These two observations may conclude that feature engineering has a much lower impact on the overall quality than the clustering algorithm. On the other hand, the selection of clustering algorithms has a limited impact on the spatial consistency of the clusters compared to feature engineering methods.

### 4.5.2 Mouse Kidney 3D Dataset

The mouse kidney 3D Mass Spectrometry Imaging dataset is primarily oriented on benchmarking methods towards high-volume scalability. It often requires additional scalability-oriented modifications to the algorithms to improve their computational complexity, like [41], yet these modifications may still be insufficient.

Indeed, spectral clustering does not allow for enough scalability, as it requires the construction of the affinity matrix describing the similarities of spectra – its size is quadratic in the number of spectra. To provide a benchmark despite this difficulty, we apply a two-step approach described by [41]. Setting the number of considered subsets to $\sqrt{n}$ where $n$ is the number of input samples, $\sqrt{n}$ elements in each subset, a two-step approach allows to effectively reduce computational complexity to $O(n\sqrt{n})$. Unfortunately, such a large number of subsets (approximately 1,168) does not lead to convergence of the two-step method. Note that the selection of another number of subsets would reduce the constant in computational complexity, not the order. At the same time, the purpose of the two-step method was to reduce the space complexity with some benefits for the speed of computations, not the reduction of the order of computational complexity. In the original work, the authors evaluate the method with up to 400,625 pixels and up to 30 subsets as this optimizes the algorithm for the space requirements.

Similarly, in the case of spatial clustering, a pairwise distance matrix is constructed, with size quadratic in the number of spectra. Unfortunately, the spatial properties of the dataset leveraged by the algorithm prohibit the application of an approach similar to the two-step method mentioned earlier. Therefore spatial clustering must be considered infeasible for the mouse kidney 3D dataset or any dataset similarly massive in the number of spectra. Moreover, it appears that the spatial clustering algorithm does not take into account the different spatial strides in the third dimension, which is caused by the data acquisition process. The two primary dimensions are constructed from a grid with $50\mu m$-by-$50\mu m$ adjacent squares by 200 averaged measurements for each square on the grid. However, the third dimension is constructed using serial tissue slices of $3.5\mu m$ thickness with multiple slices missing between them.

Due to the above reasons, only two clustering algorithms can be further evaluated: K-Means and Divisive Intelligent K-Means. In Figure 4.15 we present partitions obtained in the considered combinations of feature engineering and clustering methods. For additional clarity inspecting clusters in a 3D volume, Figure 4.16 contains each $6^{th}$ consecutive slice of the mouse kidney dataset. It can be observed that the results are mostly consistent between the computational scenarios. The most significant exception is the K-Means algorithm applied to data with no feature engineering, which does not discover any structures in the data.

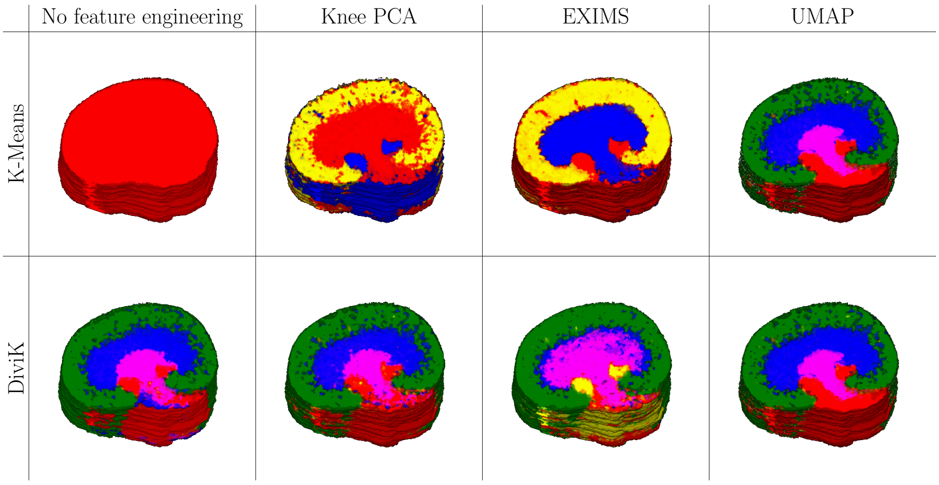


Figure 4.15: The partitions obtained with the feasible combinations of selected feature engineering and clustering methods for mouse kidney 3D data. Feature engineering methods in columns, clustering methods in rows. Presented regions were normalized in terms of colors, but no ground truth labels are propagated, as they are unspecified.

As no ground truth labels are available, the obtained regions can only be compared visually with the results already known in the literature [87, 99, 3, 2]. The detected regions share a common molecular signature across the entire

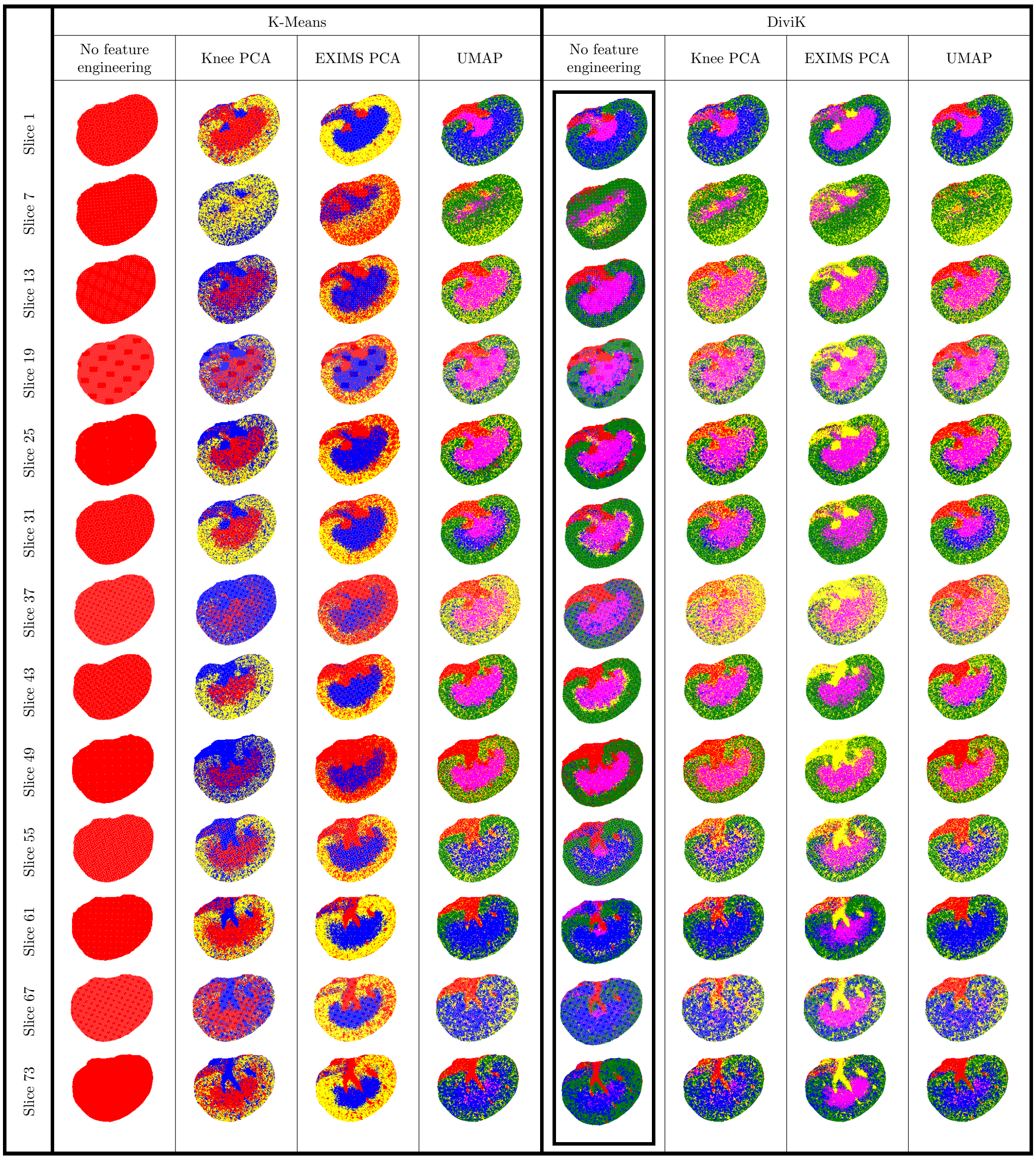


Figure 4.16: Serial slices of the partitions obtained with the feasible combinations of selected feature engineering and clustering methods for mouse kidney 3D data. Each $6^{th}$ slice of the original result cube is included in the figure. Presented regions were normalized in terms of colors, similarly to Figure 4.15.

3D volume and exhibit functional differences. Under such circumstances, we assume that both K-Means and Divisive Intelligent K-Means can lead to relevant results for high-volume data, with greater stability of the results when DiviK is applied. Therefore, we can summarize the performance of algorithms across all evaluation criteria with ranks – see Table 4.7.

Table 4.7: Rank summary of the clustering algorithms. The higher the values of a quality measure for an algorithm in Table 4.1, the lower its rank (lower rank is preferable). Scalability was assessed with a criterion, whether the algorithm accomplished computations for mouse kidney data or not.

| | Spectral clustering | K-Means | Spatial clustering | DiviK |
|---|---|---|---|---|
| Rand index ranks | 62.5 | 71 | 40.5 | **36** |
| Dice index ranks | 59 | 75 | 39 | **37** |
| EXIMS score ranks | 62.5 | **38.5** | 52.5 | 56.5 |
| scalability ranks | 72.5 | **32.5** | 72.5 | **32.5** |
| total rank | 256.5 | 217 | 204.5 | **162** |

K-Means algorithm exhibits high consistency of the obtained clusters and allows for great scalability if configured carefully. Divisive Intelligent K-Means algorithm provides high reconstruction capabilities for tumor tissue and overall tissue structure, with stable results for high-scale data, regardless of the feature engineering method. It also provides a convenient trade-off between all considered quality measures. Finally, considering the same criteria, the spatial clustering algorithm exhibits the second best total rank, probably due to its substantial synergies with other methods. However, similarly to K-Means, it requires a meticulous computational pipeline design.

Note that the DiviK approach does not require applying PCA or UMAP globally for obtaining great results with massive data. It locally optimizes feature space with GMM-based feature selection, and the hidden internal structure of the tissue may be discovered in consecutive splits. PCA and

UMAP are used in this study only as a reference. This distinction is an essential factor for computation time. With the machine equipped with 48 CPU cores and 256 GB of RAM, we computed PCA in 23.8 minutes and UMAP in 198.5 minutes. For the same dataset, the GMM-based feature filtering completes within 8.66 seconds (156 ms standard deviation). Of course, the deglomerative nature of DiviK requires multiple local filters. For this specific dataset, DiviK conducted 60 splits preceded by the filtering procedure. These repeated computations lead to 8.66 minutes of processing related to feature engineering in total.

Table 4.8: Rank summary of the feature engineering algorithms. The higher the values of a quality measure for an algorithm in Table 4.1, the lower its rank (lower rank is preferable). Scalability was assessed with a criterion, whether the algorithm accomplished computations for mouse kidney data or not.

| | UMAP | Xception | Knee PCA | EXIMS PCA | none |
|---|---|---|---|---|---|
| Rand index ranks | 48.5 | 52.5 | 49 | **25** | 35 |
| Dice index ranks | 48 | 51 | 49 | **28** | 34 |
| EXIMS score ranks | 51.5 | 32.5 | **29.5** | 60 | 36.5 |
| scalability ranks | **38** | 58 | **38** | **38** | **38** |
| total rank | 186 | 194 | 165.5 | 151 | **143.5** |

# Chapter 5

# Divisive Clustering via Variational Autoencoders

## 5.1 Variational Autoencoders

Variational Autoencoders (VAE) [70] are generative deep learning models. The two most popular applications of VAEs are:

- Generation of artificial samples from a distribution approximating the distribution of the training data set.
- Low-dimensional embedding of a data set. As opposed to methods like t-SNE [124], the obtained transformation can be applied to unseen data and is reversible.

For this work, we will focus only on the capabilities of VAEs around latent representation learning.

The main idea behind VAE is simple, although implementation details may vary and strongly influence the capability of a neural network to effectively capture all the details of training data distribution. The structure of a VAE is based on two sub-networks: encoder and decoder. The encoder is responsible for translation from original feature space to a low-dimensional latent representation, while the decoder reverses the process, reconstructing input data. The differences between original and reconstructed samples are used to update network weights.

For the latent representation to be robust, a so-called *reparametrization trick* is applied. During model training, random samples from the Gaussian noise distribution are imposed on the outputs of the encoder network before the decoding process. Without the *reparametrization trick*, the latent space lacks regularization, and the decoder model is prone to severe overfitting. Thus, *reparametrization trick* is needed to avoid poor generalization for unseen data. The architecture of such a network is presented in Figure 5.1.

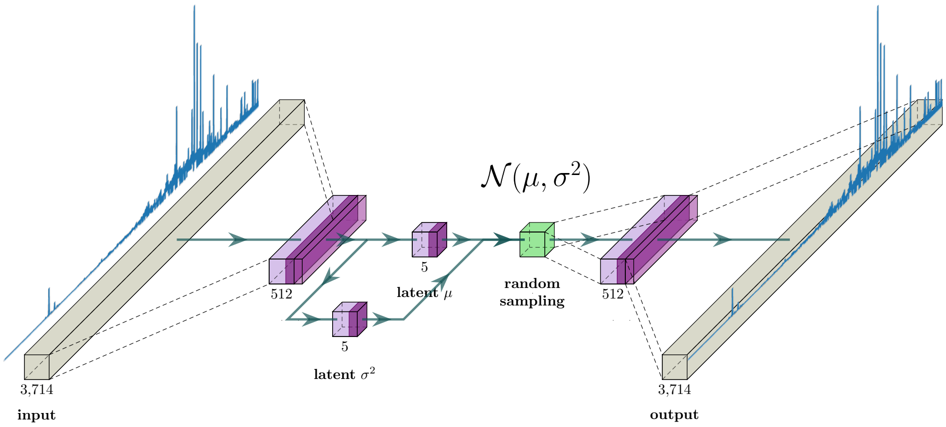


Figure 5.1: The architecture of the Variational Autoencoder used for processing MSI data as proposed by [2]. The input size is fixed for a specific dataset and the same as the output size. After the two non-linear dimensionality reductions happen in the encoder part, a random sampling occurs – this is the reparametrization trick (green box in the middle). The reparametrized latent representation is then decoded back to the original feature space.

To proceed with fitting a VAE model to the data, one needs to define its loss function. The original publication [70] introduces the variational lower bound on the marginal likelihood of a VAE model, which could suggest what kinds of loss functions would optimize the parameters of the two sub-networks. In the Equation 5.1 one can see, how both the *reparametrization trick* and reconstruction effectiveness contribute to model relevance:

$$\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) = -D_{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})\Big) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\Big[\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})\Big] \tag{5.1}$$

where:

$\phi$ encoder model parameters;

$\theta$ decoder model parameters;

$\mathbf{z}$ latent variable (in the encoded space);

$\mathbf{x}$ training data (in the decoded space);

$\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$ the variational lower bound on the marginal likelihood of a VAE model;

$q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$ approximate distribution at the output of the encoder model;

$p_\theta(\mathbf{z})$ (Gaussian) prior distribution of a latent variable $\mathbf{z}$;

$D_{KL}$ Kullback-Leibler divergence – in this case between approximate latent representation distribution and the standard Gaussian distribution;

$\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})$ log likelihood of the conditional distribution of decoder outputs;

$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}$ expected value of the log likelihood of the decoder outputs distribution – often expressed in terms of reconstruction error and specifically the square Euclidean distance between original and reconstructed vector.

The likelihood maximization problem expressed by Equation 5.1 can be solved for parameters $\phi$ and $\theta$ using stochastic optimization methods like SGD or Adagrad [45].

There are variations of VAE which may take into account a target variable when constructing the latent space (like conditional VAE [115]) or implement more than one latent space [121, 101]. One can also find many domain-optimized VAE implementations, e.g., VASC for single-cell RNA sequencing (scRNA-Seq) [131] that uses an additional zero-inflated layer with Gumbel distribution for the reconstruction of zeroes dominating scRNA-Seq datasets or fully-connected VAE for MSI [2].

## 5.2 Autoencoders for Mass Spectrometry Imaging Data

The trend for applying Variational Autoencoders for Mass Spectrometry Imaging data is definitely rising, which could be observed with the number of various preprints and other works appearing in the field. However, since they have not yet been peer-reviewed, we will omit them in this dissertation and focus on two reviewed approaches.

The first approach featuring an Autoencoder for Mass Spectrometry Imaging data used a 2D (409x404) image of a mouse brain with 7036 intensities each [117]. Note, that the *reparametrization trick* is missing as it is not a *Variational* Autoencoder. The authors applied a zero-filling procedure to create a consistent $m/z$ axis, a Savitzky-Golay filter to reduce the noise, and a peak-picking method based on the noise level. After these steps, all spectra were normalized to a consistent total ion count. The Autoencoder had 7036 input nodes, 15 hidden nodes, and 7036 output nodes. The training was conducted purely unsupervised and compared to PCA and Non-Negative Matrix Factorization (NNMF). The authors presented that the $5^{th}$ PCA component already accounted for just 0.44% of the variance, yet the information loss across the obtained 15 components was significant in biological relevancy. At the same time, spatial maps of the representation constructed with NNMF provided a very low contrast. Authors argued about the normalization opportunities but indicated problems related to single-pixel noises dominating the image. They indicated that by the definition of NNMF, many features obtained this way represent noise in the data. Similarly to NNMF, Autoencoder generated few uninformative features from the data – which is natural for the non-variational character of this method. However, the Autoencoder compressed much of the molecular heterogeneity into a few informative features. These few informative features capture almost all structural features and regions of the mouse brain.

Since that time, a new approach has been proposed by [2]. In Figure 5.1 we present the exact architecture of the network used. The method is shown to work well for raw and unprocessed spectra, which are at its input. The original spectrum is reduced through 512 fully connected hidden neurons followed by batch normalization [62] and ReLU activation function. The latent representation is obtained with the next layer of 5 fully connected neurons. Then the random sampling occurs (Figure 5.1, in green), and the decoder part of the network reconstructs the reparametrized spectrum. It consists of 512 fully connected hidden neurons followed by batch normalization and ReLU, and the fully connected layer with the number of neurons equal to the number of input dimensions.

Such architecture has two remarkable properties:

- Fully connected neural network introduces a massive number of parameters. Thus the model needs to be relatively shallow and easily becomes overparameterized. Although in traditional learning theory, this often leads to overfitting, in neural networks, it may be beneficial for both optimization, and generalization [13].

- To make transfer learning of the obtained embedding possible, two (or more) MSI studies require the same mass-to-charge ratio axis. Although useful for analysis of a single MSI dataset, possible reuse of the pretrained encoder model would require dedicated experiment scenario planning far ahead of data acquisition. Such long-term planning is unlikely for basic real-life scenarios. However, it exhibits high potential when building an extensive database focused on a specific biological phenomenon.

Recently, this work has been extended to a classification case as well [4]. Authors use pretrained VAE to provide a consistent embedding for a much larger set of data and train a classifier on top of the obtained embedding. Despite providing better precision, recall, accuracy, and F1 score, the solution was also approximately 174 times faster than the traditional Support Vector Machine.

Interestingly, Figure 2 in the publication [4] indicates the application of batch normalization together with dropout, which may often be a sub-optimal approach [73]. However, from the discussion with the authors, we know that they actually observed improved outcomes in this specific configuration.

## 5.3 Methods

Inspired by the promising results of [2] and further [4], we reuse the VAE approach as a part of the DiviK framework. This combination leads us to the *DiVAE* method: Divisive clustering with Variational Autoencoder, which is further described below. Its formulation is consistent with the original DiviK idea for clustering with local optimization of feature space, but the components inside are redefined.

### 5.3.1 Local Feature Space Optimization

To obtain the locally optimized feature space for clustering purposes, we apply Variational Autoencoder as described by [2]. The exact same architecture is used (see Figure 5.1): 512 intermediate features and 5 latent ion images.

In our case, the difference is that the input features in the considered dataset are already mathematically modeled and not a set of unprocessed mass channels. Due to that fact, we can use a batch size of 8192 spectra for training the VAE model.

The neural network is trained for 500 epochs with a learning rate schedule, rising from 0.001 to 0.01 in the first half of the training and then decaying back to 0.001.

### 5.3.2 Clustering Method

The selection of a clustering method that supports the expected data distribution is straightforward in the DiVAE algorithm. The Equation 5.1 imposes an optimization goal for the VAE to distribute encoder outputs in a way that follows Gaussian distribution. Moreover, the sampling occurring during the reparametrization trick also draws from the Gaussian distribution. Therefore, Gaussian Mixture Model seems the best choice for clustering the data embedded with the VAE.

Gaussian Mixture Model in a non-Bayesian formulation requires the number of clusters specified upfront. We follow a similar procedure as with the K-Means algorithm in DiviK: we sweep through the set of possible cluster numbers and evaluate the obtained model based on some unsupervised quality criteria. For that purpose, we use classical Bayesian Information Criterion (BIC) [109]. We calculate Youden's J statistic [137] to identify the number of clusters *good enough* in terms of BIC.

### 5.3.3 Stop Condition

Stop condition does not require significant changes compared to Divisive Intelligent K-Means, although confirmation of subregion heterogeneity can be considerably simplified. With the K-Means algorithm, the research community considers a single cluster case situation unusual enough for most clustering quality indices to omit this case. Fortunately, with Gaussian Mixture Model, a single-component case is taken into account, and heterogeneity confirmation becomes just a part of the clustering process.

## 5.4 Experimental Settings

We reused the subset of experimental settings from Section 4.4, which was relevant for the OSCC dataset. The original publication [2] already validated scalability with the mouse kidney 3D dataset for all the algorithms we use as components of the DiVAE algorithm. Repeating the computations multiple times with the feasible order of computational complexity in each iteration does not affect the overall method feasibility. Hence, this work focuses mainly on the biological relevance of obtained results for OSCC data.

To simplify the comparison, we selected the three top combinations of feature engineering and clustering methods in terms of *overall quality* from the study described in Chapter 4. Note that this also includes the top combinations in terms of Dice index, relative EXIMS score, and adjusted Rand index.

The neural network is trained for 500 epochs with a learning rate schedule, rising from 0.001 to 0.01 in the first half of the training and then decaying back to 0.001.

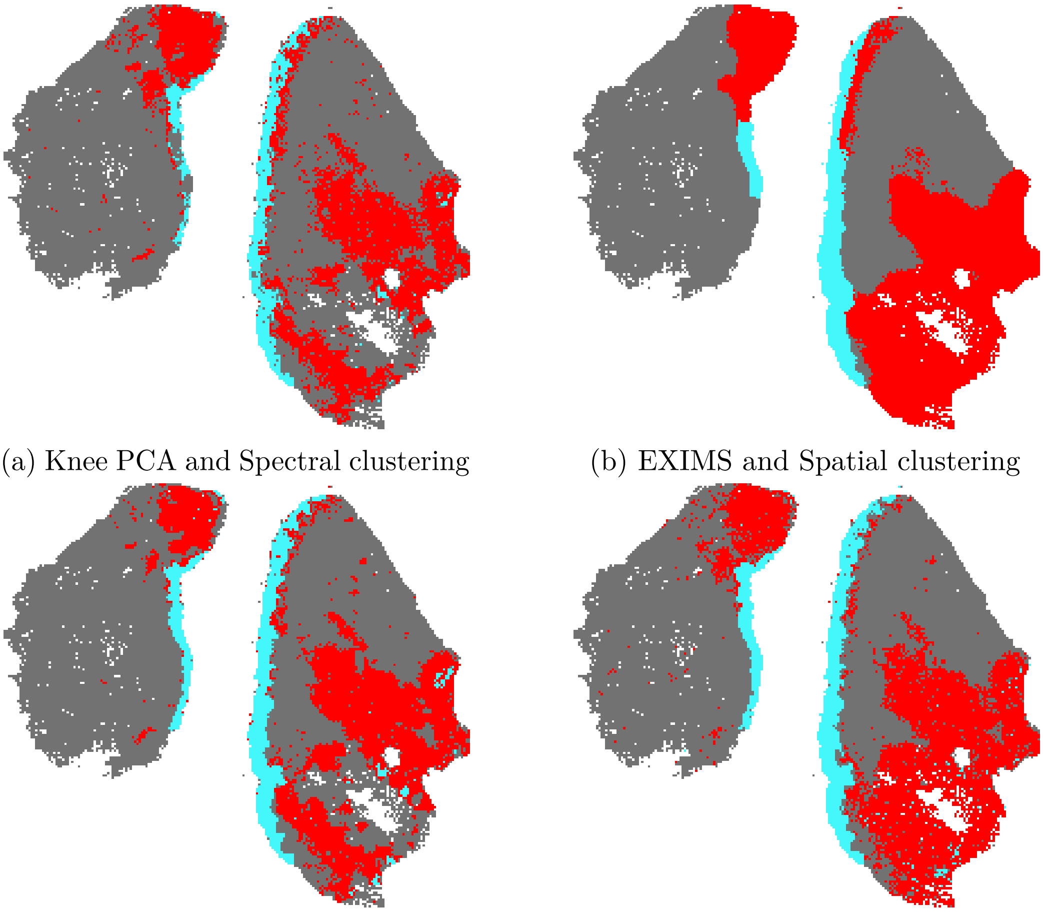
Since the Variational Autoencoder operates with high dimensional data or even unprocessed MSI spectra, we do not combine it with other feature engineering methods, as it would lead to severe overfitting of the obtained deep learning model.

To evaluate the results quantitatively, we are using the annotation translation process described in Section 4.4.1.

## 5.5 Results and Discussion

The clustering results of all considered configurations are presented in Figure 5.2. DiVAE provides results visually consistent with a baseline obtained with known approaches, although it covers the tumor region much more precisely. This is confirmed with Dice index computed for tumor region (see Table 5.1).

Spatial clustering with EXIMS PCA still provides the top Dice index and adjusted Rand index. At the same time, the capability to reconstruct overall tissue composition (expressed with Rand index) and tumor region (expressed with Dice index) by the DiVAE algorithm is similar. It yields the second top Dice index and the second top adjusted Rand index. In both cases, it is substantially higher than the remaining algorithms' results. Furthermore, DiVAE yields the highest relative EXIMS score. Note that the relative EXIMS score values differ from the ones presented in Section 4.5.1 and Table 4.1, as the results from the DiVAE method yield a higher absolute value of the EXIMS score than already observed. Finally, the *overall quality* $d(0,0,0)$



(a) Knee PCA and Spectral clustering

(b) EXIMS and Spatial clustering

(c) DiviK

(d) DiVAE

Figure 5.2: The partitions obtained with the combinations of feature engineering and clustering methods for OSCC data as described in Section 5.4. Presented regions correspond to ROIs defined by pathologist: red – tumor, cyan – healthy epithelium, gray – other tissue (compare Figure 2.6). The cluster annotations were obtained with the label translation process described in Section 4.4.1.

and *overall quality* $d(1,1,1)$ of the DiVAE method is substantially improved compared to the DiviK and other methods (see Table 5.2).

In Figure 5.3 we compare the partition obtained in the first step of deglomerative approaches (DiviK and DiVAE). In the first step of DiviK, the healthy epithelium is clustered together with the tumor on one sample (Figure 5.3a). These regions get separated in further analysis. Conversely,

Table 5.1: Basic clustering quality measures computed for OSCC data. The preferred value in the column is **bold**.

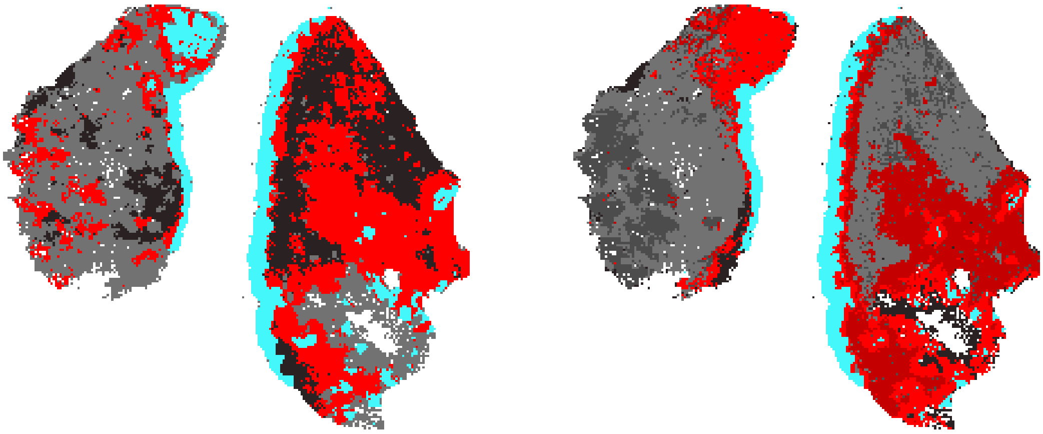
| clustering algorithm | global feature engineering method | adjusted Rand index | Dice index | relative EXIMS score |
|---|---|---|---|---|
| Spectral | Knee PCA | 0.4594 | 0.6897 | 0.9422 |
| Spatial | EXIMS PCA | **0.7035** | **0.8672** | 0.6646 |
| DiviK | none | 0.5433 | 0.7372 | 0.9525 |
| DiVAE | none | 0.6731 | 0.8505 | **1.0000** |

Table 5.2: Summarizing clustering quality measures computed for OSCC data. The preferred value in the column is **bold**.

| clustering algorithm | global feature engineering method | overall quality $d(0,0,0)$ | overall quality $d(1,1,1)$ |
|---|---|---|---|
| Spectral | Knee PCA | 1.2547 | 0.6260 |
| Spatial | EXIMS PCA | 1.2995 | 0.4669 |
| DiviK | none | 1.3213 | 0.5291 |
| DiVAE | none | **1.4753** | **0.3595** |

the first step of DiVAE provides a distinction between tumor and epithelium region. Therefore, DiVAE requires even fewer steps to achieve a similar result. Moreover, upon visual comparison, the DiVAE clusters seem to correspond more to the actual regions defined by the pathologist than the DiviK ones.

We analyze the latent space obtained from the OSCC dataset embedding on the top level. As embedding each spectrum in the MSI dataset does not change its dual nature, we visualize the spatial distribution of the latent variables in Figure 5.4. We use the contrast enhancement method classical for MSI data [98] to indicate the most vital differences. It is clearly visible that the tumor region differentiates from healthy epithelium (Figure 5.4c and 5.4e). At the same time, the epithelial origin of the OSCC tumor can be spotted through the similarities in Figure 5.4b. Finally, some minor differences

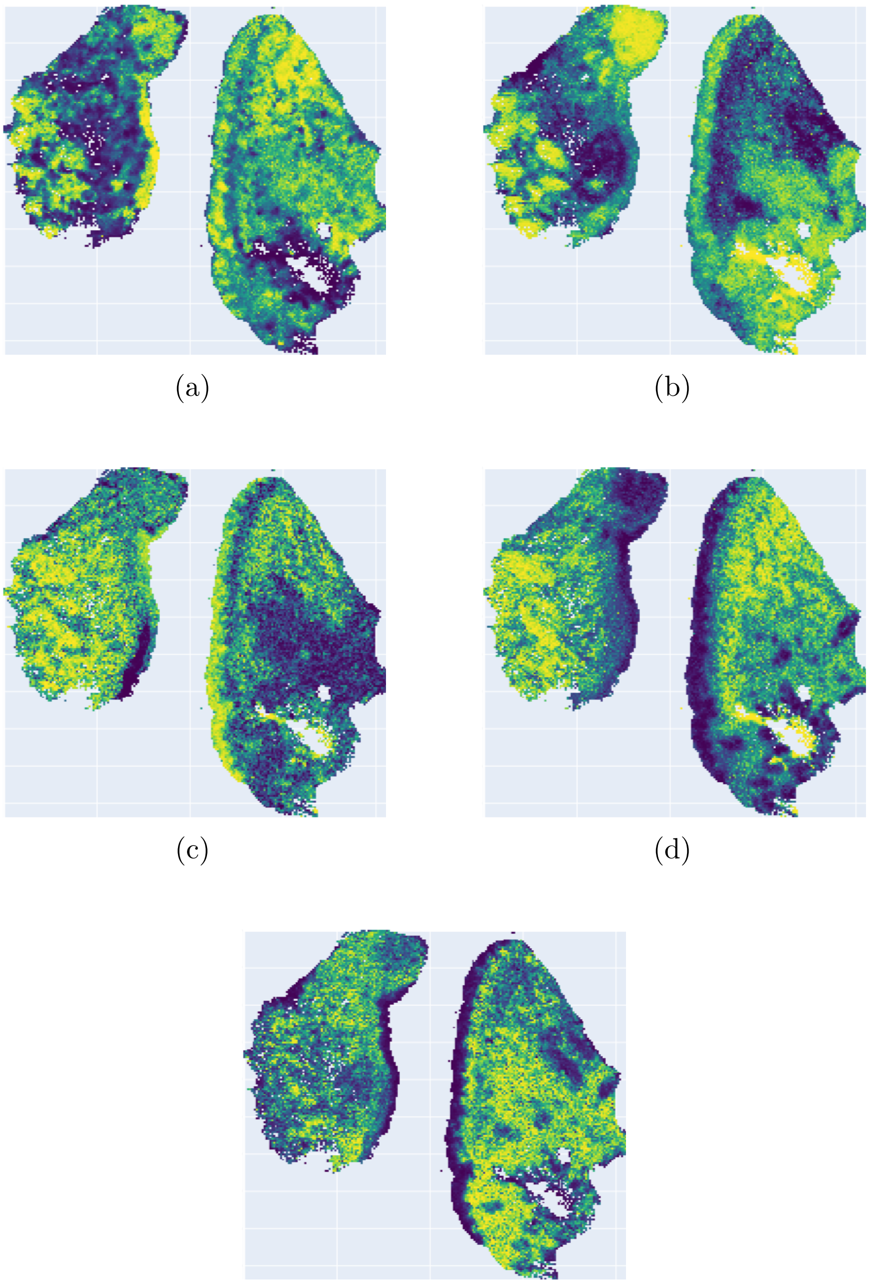


(a) DiviK

(b) DiVAE

Figure 5.3: The partitions obtained in the first step with hierarchical methods for OSCC data in DiVAE experiments. Presented regions correspond to ROIs defined by pathologist: red – tumor, cyan – healthy epithelium, gray – other tissue (compare Figure 2.6). The cluster annotations were obtained with the label translation process described in Section 4.4.1, but separate clusters have been marked with different shades for visual comparison.

between tumor tissue coming from different patients can be observed in Figure 5.4e and 5.4d.



(a)

(b)

(c)

(d)

(e)

Figure 5.4: Latent ion images obtained for the OSCC dataset. One can observe that the biological structures appear in this representation. The molecular patterns characteristic of visible structures dominate the dataset and limit the possibility of detailed analysis.

# Chapter 6

# Summary

## 6.1 Thesis Summary

The results presented in this dissertation justify the theses presented in Section 1.1. Section 4.4.2 explains how the biological relevance of the obtained results could be assessed numerically. A few classical quality measures are presented, and a few more are derived from investigating the trade-off between them. Sections 4.5.1 and 5.5 present results of two proposed stepwise approaches using these numerical quality measures for obtained segmentations of Oral Squamous Cell Carcinoma full-tissue with both DiviK and DiVAE. These results confirm thesis 1. Thesis 2 was proven in Section 4.5.2, where DiviK and K-Means algorithms were the only ones to accomplish the clustering task for 3D mouse kidney MSI data. Finally, thesis 3 is confirmed in Chapter 5 that formulates and evaluates the DiVAE algorithm with the OSCC dataset against the state-of-the-art methods in the domain.

We proposed the methodology that builds upon the experiences of numerous data processing experts and the characteristics of big -omics data gathered in Chapter 3. The stepwise approach is at the core of the framework we propose, with feature engineering and clustering alternating when processing finer and finer details. The classical K-Means algorithm was adjusted for clustering MSI data and combined with the GMM-based feature selection method. As the literature suggests, a dedicated calibration step was required to adjust the K-Means algorithm for the data characteristics. We analyzed the quality of obtained segmentations numerically and provided a few kinds of summaries that assess the trade-off between basic quality measures from different perspectives, which was suggested by the experts in the field reviewing our journal contribution. We checked the relevance of the numerical evaluation via effect size.

Furthermore, we ran large-scale computations with the DiviK framework to ensure it could process high-volume data. Finally, this work presents the DiviK framework as a flexible concept when introducing DiVAE. This allows for taking advantage of DiviK's stepwise approach and the newest techniques in the field, like deep learning.

## 6.2 Potential Applications

Divisive Intelligent K-Means framework is not limited only to tumor delineation in Mass Spectrometry Imaging data. There are a few more areas in which its usefulness is investigated. As the purpose of this section is primarily to show the opportunities (and opportunities as such may not be fully validated yet), I will also refer to some unpublished work taking place in our department.

One of the recent (unpublished) works based on single-cell RNA Sequencing uses DiviK as an alternative clustering method capable of detecting small clusters in the data. When applied to a well-known immunity-related dataset [59], it provides high coverage in terms of weighted Dice index compared to the original clustering by the Seurat algorithm – from 88.15% to 93.64% (no ground-truth labels available). Obtained clusters are further used for downstream analysis, including effect size profiling, biomarker identification, and quantitative results processing,

Similarly, mass cytometry data analysis can be used to analyze antibodies present in the organism and to figure out the background behind tuberculosis drug resistance. Unfortunately, mass cytometry datasets often span millions of observations, which makes most sophisticated clustering algorithms infeasible. DiviK's scalability can be used to identify small groups in a dataset of similar volume. Such groups could provide insights into the drug resistance antibody signature for tuberculosis.

Another researcher uses DiviK to conduct quality control of the obtained Mass Spectrometry Imaging dataset. The dataset consists of numerous small circular tissue sections. The only annotations provided by the pathologist were assigned to tissue samples, not individual mass spectra. DiviK is used there for anomaly detection – to identify situations with conflicting molecular information. These situations often translate to a discovery of tumor cells in the sample annotated as healthy. Selected cases may be directed to review and result in a curated label set. Other cases where the proportion between healthy and tumorous areas is similar could be removed from the training dataset. Beyond the fact that it can be used for training set selection, it addresses a relatively trendy area nowadays: *data-centric AI*.

Therefore, DiviK could be applied for curating the labels obtained in the classical annotation process. In biomedical data, this has a vast potential, as most of the annotation is based on phenotype, while various *-omics* approaches provide more detailed information. A great example is facioscapulohumeral muscular dystrophy (FSHD), which may be caused by two different genetic backgrounds (type I and II), and only a multi-step genomics data analysis (like DiviK's one) can discover the difference. A similar scenario occurs for thyroid cancer [94]: some subtypes of thyroid cancer cannot be visually discerned even upon resection. A multi-stage analysis with DiviK allows to compare protein profiles and assess similarity across numerous tissue samples.

## 6.3 Future Directions

There are many ways this work could be continued, with two notable directions being algorithm improvements and the spectrum of applications for different kinds of data. As the latter was partly addressed in Section 6.2, here I will focus only on the algorithmic frontiers potentially worth investigating.

An exciting opportunity would be to refine the definition of the GAP index. The authors suggest a research direction by sampling the reference dataset from the cluster limits instead of the entire dataset [119]. As this idea was not yet evaluated for MSI data, one could argue what would be the influence of such a redefinition of the GAP index on the heterogeneity confirmation. At the same time, it is a promising solution for speeding up computations of the GAP index. On the implementation level, instead of using uninitialized K-Means for computing the partition and dispersions afterward, one could reuse the centroids of the obtained clusters. Hypothetically, this would significantly decrease the time required for K-Means convergence over the sample unless serious skewness is present in the real data distribution. Since extensive time profiling indicated GAP index calculation as a primary source of computation time, such modification could have a strong positive effect on the overall algorithm performance.

Secondly, Chapter 5 shows the usefulness of applying Deep Neural Networks to MSI data. The Neural Network architecture used in this work is exactly the same as in [2]. However, literature in the adjacent biological domains like scRNA-Seq or mass cytometry shows us that there is a considerable space for architectural optimizations [131, 118, 50]. Different architectures are proposed to address numerous challenges in the scRNA-Seq data, like a high proportion of zeros observed in the dataset. Such phenomena are addressed with a dedicated architecture design, including elements like zero-inflation layer, residual connections, atrous convolutions, and others, depending on the details in the process of data acquisition. While this work demonstrated the possibility of using Neural Networks and the DiviK framework in a single processing pipeline, the area is heavily underutilized.

Another great inspiration could be the domain on Natural Language Processing, with the attention mechanism proposed recently [127] and transformer models [40, 75, 100, 35]. There are initial successes in applying transformer models to other domains like computer vision [43, 30], or audio processing [16]. One of the major advantages these solutions offer is the unsupervised pre-training capability, which could use the massive MSI datasets without annotation. Hypothetically, this could provide even better robustness against most of the effects manually removed during the MSI dataset preprocessing (see Chapter 2). Indeed, the first work already appeared in the area of Mass Spectrometry [112] and presented the transformer method MassGenie as extremely effective in predicting the 2D structure of the molecule. Unfortunately, this area has not yet been tackled in Mass Spectrometry Imaging data analysis.

# Chapter 7

# Acknowledgment

I am very thankful to the Institute of Oncology in Gliwice for possibility of working with Mass Spectrometry Imaging datasets and their help in interpretation of results, especially:

- prof. dr hab. Piotr Widłak
- dr hab. Monika Pietrowska, prof. nzw. NIO
- dr Marta Gawin
- dr n. med. Mykola Chekan

Last, but not least, I would like to thank my supervisor, prof. Joanna Polańska, for her endless patience and support.

This study has been financially supported by NCN grant BITIMS no. 2015/19/B/ST6/01736 and NCBiR grant AIDA no. I029/17-POWR.03.02.00-IP.08-00-DOK/17. Computations were carried out using the GeCONiI infrastructure (grant POIG.02.03.01 24 099).

# Bibliography

[1] Metaspace annotation platform: datasets summary. `https://metaspace2020.eu/datasets/summary`. Accessed: 2022-05-02.

[2] Walid M Abdelmoula, Begona Gimenez-Cassina Lopez, Elizabeth C Randall, Tina Kapur, Jann N Sarkaria, Forest M White, Jeffrey N Agar, William M Wells, and Nathalie YR Agar. Peak learning of mass spectrometry imaging data using artificial neural networks. *Nature communications*, 12(1):1–13, 2021.

[3] Walid M Abdelmoula, Nicola Pezzotti, Thomas Hölt, Jouke Dijkstra, Anna Vilanova, Liam A McDonnell, and Boudewijn PF Lelieveldt. Interactive visual exploration of 3d mass spectrometry imaging data using hierarchical stochastic neighbor embedding reveals spatiomolecular structures at full data resolution. *Journal of proteome research*, 17(3):1054–1064, 2018.

[4] Walid M Abdelmoula, Sylwia A Stopka, Elizabeth C Randall, Michael Regan, Jeffrey N Agar, Jann N Sarkaria, William M Wells, Tina Kapur, and Nathalie YR Agar. massnet: integrated processing and classification of spatially resolved mass spectrometry data using deep learning for rapid tumor delineation. *Bioinformatics*, 38(7):2015–2021, 2022.

[5] Charu C Aggarwal, Joel L Wolf, Philip S Yu, Cecilia Procopiuc, and Jong Soo Park. Fast algorithms for projected clustering. *ACM SIGMoD Record*, 28(2):61–72, 1999.

[6] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11(1):5–33, 2005.

[7] Hongryul Ahn, Inuk Jung, Heejoon Chae, Minsik Oh, Inyoung Kim, and Sun Kim. Idea: Integrating divisive and ensemble-agglomerate hierarchical clustering framework for arbitrary shape data. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 2791–2800. IEEE, 2021.

[8] Michaela Aichler and Axel Walch. Maldi imaging mass spectrometry: current frontiers and perspectives in pathology research and practice. *Laboratory investigation*, 95(4):422–431, 2015.

[9] Theodore Alexandrov. Spatial metabolomics and imaging mass spectrometry in the age of artificial intelligence. *Annual review of biomedical data science*, 3:61–87, 2020.

[10] Theodore Alexandrov, Michael Becker, Sören-oliver Deininger, Gunther Ernst, Liane Wehder, Markus Grasmair, Ferdinand von Eggeling, Herbert Thiele, and Peter Maass. Spatial segmentation of imaging mass spectrometry data with edge-preserving image denoising and clustering. *Journal of proteome research*, 9(12):6535–6546, 2010.

[11] 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.

[12] Theodore Alexandrov and Jan Hendrik Kobarg. Efficient spatial segmentation of large imaging mass spectrometry datasets with spatially aware clustering. *Bioinformatics*, 27(13):i230–i238, 2011.

[13] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019.

[14] Farzana Anowar, Samira Sadaoui, and Bassant Selim. Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). *Computer Science Review*, 40:100378, 2021.

[15] Anestis Antoniadis, Jérémie Bigot, and Sophie Lambert-Lacroix. Peaks detection and alignment for mass spectrometry data. *Journal de la Société Française de Statistique*, 151(1):17–37, 2010.

[16] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

[17] Jens Behrmann, Christian Etmann, Tobias Boskamp, Rita Casadonte, Jörg Kriegsmann, and Peter Maa$\beta$. Deep learning for tumor classification in imaging mass spectrometry. *Bioinformatics*, 34(7):1215–1223, 2018.

[18] James C Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

[19] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[20] Christian Bohm, K Railing, H-P Kriegel, and Peer Kroger. Density connected clustering with local subspace preferences. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 27–34. IEEE, 2004.

[21] Daniel Boley. Principal direction divisive partitioning. *Data mining and knowledge discovery*, 2(4):325–344, 1998.

[22] Nadia Bolshakova and Francisco Azuaje. Cluster validation techniques for genome expression data. *Signal processing*, 83(4):825–833, 2003.

[23] Charles Bouveyron, Stéphane Girard, and Cordelia Schmid. High-dimensional data clustering. *Computational Statistics & Data Analysis*, 52(1):502–519, 2007.

[24] Paul S Bradley and Usama M Fayyad. Refining initial points for k-means clustering. In *ICML*, volume 98, pages 91–99. Citeseer, 1998.

[25] F Bray, J Ferlay, I Soerjomataram, RL Siegel, LA Torre, and A Jemal. Erratum: Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *Ca Cancer J Clin*, 70(4):313, 2020.

[26] Michael D Buck, Ryan T Sowell, Susan M Kaech, and Erika L Pearce. Metabolic instruction of immunity. *Cell*, 169(4):570–586, 2017.

[27] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[28] Yaoming Cai, Meng Zeng, Zhihua Cai, Xiaobo Liu, and Zijia Zhang. Graph regularized residual subspace clustering network for hyperspectral image clustering. *Information Sciences*, 578:85–101, 2021.

[29] Joel Luis Carbonera and Mara Abel. An entropy-based subspace clustering algorithm for categorical data. In *2014 IEEE 26th international conference on tools with artificial intelligence*, pages 272–277. IEEE, 2014.

[30] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[31] M Emre Celebi, Hassan A Kingravi, and Patricio A Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert systems with applications*, 40(1):200–210, 2013.

[32] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1):89–97, 2004.

[33] Mark Ming-Tso Chiang and Boris Mirkin. Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. *Journal of classification*, 27(1):3–40, 2010.

[34] Laura QM Chow. Head and neck cancer. *New England Journal of Medicine*, 382(1):60–72, 2020.

[35] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[36] Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 118–127, 2016.

[37] Renato Cordeiro De Amorim and Boris Mirkin. Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. *Pattern Recognition*, 45(3):1061–1075, 2012.

[38] 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.

[39] Soren-Oliver Deininger, Matthias P Ebert, Arne Futterer, Marc Gerhard, and Christoph Rocken. Maldi imaging combined with hierarchical clustering as a new tool for the interpretation of complex human cancers. *Journal of proteome research*, 7(12):5230–5236, 2008.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[41] Alex Dexter, Alan M Race, Rory T Steven, Jennifer R Barnes, Heather Hulme, Richard JA Goodwin, Iain B Styles, and Josephine Bunch. Two-phase and graph-based clustering methods for accurate and efficient segmentation of large mass spectrometry images. *Analytical chemistry*, 89(21):11293–11300, 2017.

[42] Alex Dexter, Spencer A Thomas, Rory T Steven, Kenneth N Robinson, Adam J Taylor, Efstathios Elia, Chelsea Nikula, Andrew D Campbell, Yulia Panina, Arafath K Najumudeen, et al. Training a neural network to learn other dimensionality reduction removes data size restrictions in bioinformatics and provides a new route to exploring data representations. *bioRxiv*, 2020.

[43] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[44] Geoff M Downs and John M Barnard. Clustering methods and their uses in computational chemistry. *Reviews in computational chemistry*, 18:1–40, 2003.

[45] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[46] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of cybernetics*, 3:32–57, 1973.

[47] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[48] Sabine A Eming, Thomas A Wynn, and Paul Martin. Inflammation and metabolism in tissue repair and regeneration. *Science*, 356(6342):1026–1030, 2017.

[49] Pedro Escoll and Carmen Buchrieser. Metabolic reprogramming of host cells upon bacterial infection: Why shift to a warburg-like metabolism? *The FEBS journal*, 285(12):2146–2160, 2018.

[50] Mario Flores, Zhentao Liu, Tinghe Zhang, Md Musaddaqui Hasib, Yu-Chiao Chiu, Zhenqing Ye, Karla Paniagua, Sumin Jo, Jianqiu Zhang, Shou-Jiang Gao, et al. Deep learning tackles single-cell analysis—a survey of deep learning for scrna-seq analysis. *Briefings in Bioinformatics*, 23(1):bbab531, 2022.

[51] Timo Gaber, Cindy Strehl, and Frank Buttgereit. Metabolic regulation of inflammation. *Nature Reviews Rheumatology*, 13(5):267–279, 2017.

[52] Wil Gardner, Suzanne M Cutts, Benjamin W Muir, Robert T Jones, and Paul J Pigram. Visualizing tof-sims hyperspectral imaging data using color-tagged toroidal self-organizing maps. *Analytical chemistry*, 91(21):13855–13865, 2019.

[53] Markus Grasmair. Locally adaptive total variation regularization. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 331–342. Springer, 2009.

[54] Nizar Grira, Michel Crucianu, and Nozha Boujemaa. Unsupervised and semi-supervised clustering: a brief survey. *A review of machine learning techniques for processing multimedia content*, 1:9–16, 2004.

[55] Dan Guo, Melanie Christine Föll, Veronika Volkmann, Kathrin Enderle-Ammour, Peter Bronsert, Oliver Schilling, and Olga Vitek. Deep multiple instance learning classifies subtissue locations in mass spectrometry images from tissue-level annotations. *Bioinformatics*, 36(Supplement_1):i300–i308, 2020.

[56] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of intelligent information systems*, 17(2):107–145, 2001.

[57] Greg Hamerly and Charles Elkan. Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 600–607, 2002.

[58] Katsuji Hattori, Mayumi Kajimura, Takako Hishiki, Tsuyoshi Nakanishi, Akiko Kubo, Yoshiko Nagahata, Mitsuyo Ohmura, Ayako Yachie-Kinoshita, Tomomi Matsuura, Takayuki Morikawa, et al. Paradoxical atp elevation in ischemic penumbra revealed by quantitative imaging mass spectrometry, 2010.

[59] Hugo G Hilton, Nimrod D Rubinstein, Peter Janki, Andrea T Ireland, Nicholas Bernstein, Nicole L Fong, Kevin M Wright, Megan Smith, David Finkle, Baby Martin-McNulty, et al. Single-cell transcriptomics of the naked mole-rat reveals unexpected features of mammalian immunity. *PLoS biology*, 17(11):e3000528, 2019.

[60] Paolo Inglese, Gonçalo Correia, Zoltan Takats, Jeremy K Nicholson, and Robert C Glen. Sputnik: an r package for filtering of spatially related peaks in mass spectrometry imaging data. *Bioinformatics*, 35(1):178–180, 2019.

[61] Paolo Inglese, James S McKenzie, Anna Mroz, James Kinross, Kirill Veselkov, Elaine Holmes, Zoltan Takats, Jeremy K Nicholson, and Robert C Glen. Deep learning and 3d-desi imaging reveal the hidden metabolic heterogeneity of cancer. *Chemical science*, 8(5):3500–3511, 2017.

[62] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[63] Alessio Ishizaka, Banu Lokman, and Menelaos Tasiou. A stochastic multi-criteria divisive hierarchical clustering algorithm. *Omega*, 103:102370, 2021.

[64] Anil K Jain and Richard C Dubes. *Algorithms for clustering data.* Prentice-Hall, Inc., 1988.

[65] Emrys A Jones, Sören-Oliver Deininger, Pancras CW Hogendoorn, André M Deelder, and Liam A McDonnell. Imaging mass spectrometry statistical analysis. *Journal of proteomics*, 75(16):4962–4989, 2012.

[66] Emrys A Jones, Alexandra van Remoortere, René JM van Zeijl, Pancras CW Hogendoorn, Judith VMG Bovée, André M Deelder, and Liam A McDonnell. Multiple statistical analysis techniques corroborate intratumor heterogeneity in imaging mass spectrometry datasets of myxofibrosarcoma. *PloS one*, 6(9):e24913, 2011.

[67] Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. Density-connected subspace clustering for high-dimensional data. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 246–256. SIAM, 2004.

[68] Leonard Kaufman and Peter J Rousseeuw. Partitioning around medoids (program pam). *Finding groups in data: an introduction to cluster analysis*, 344:68–125, 1990.

[69] Yeongwoo Kim, Ezeddin Al Hakim, Johan Haraldson, Henrik Eriksson, José Mairton B da Silva, and Carlo Fischione. Dynamic clustering in federated learning. In *ICC 2021-IEEE International Conference on Communications*, pages 1–6. IEEE, 2021.

[70] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[71] Agata Kurczyk, Marta Gawin, Mykola Chekan, Agata Wilk, Krzysztof Łakomiec, Grzegorz Mrukwa, Katarzyna Frątczak, Joanna Polanska, Krzysztof Fujarewicz, Monika Pietrowska, et al. Classification of thyroid

tumors based on mass spectrometry imaging of tissue microarrays; a single-pixel approach. *International Journal of Molecular Sciences*, 21(17):6289, 2020.

[72] Hubert Lawrence and Arabie Phipps. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[73] Xiang Li, Shuo Chen, Xiaolin Hu, and Jian Yang. Understanding the disharmony between dropout and batch normalization by variance shift. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2682–2690, 2019.

[74] John Lipor and Laura Balzano. Clustering quality metrics for subspace clustering. *Pattern Recognition*, page 107328, 2020.

[75] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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

[77] Chao Lu and Craig B Thompson. Metabolic regulation of epigenetics. *Cell metabolism*, 16(1):9–17, 2012.

[78] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[79] Veronica Mainini, Maciej Lalowski, Athanasios Gotsopoulos, Vasiliki Bitsika, Marc Baumann, and Fulvio Magni. Maldi-imaging mass spectrometry on tissues. In *Clinical Proteomics*, pages 139–164. Springer, 2015.

[80] Michal Marczyk, Roman Jaksik, Andrzej Polanski, and Joanna Polanska. Adaptive filtering of microarray gene expression data based on gaussian mixture decomposition. *BMC bioinformatics*, 14(1):101, 2013.

[81] Michal Marczyk, Roman Jaksik, Andrzej Polanski, and Joanna Polanska. Gamred—adaptive filtering of high-throughput biological data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(1):149–157, 2018.

[82] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[83] Daisuke Miura, Yoshinori Fujimura, Mayumi Yamato, Fuminori Hyodo, Hideo Utsumi, Hirofumi Tachibana, and Hiroyuki Wariishi. Ultrahighly sensitive in situ metabolomic imaging for visualizing spatiotemporal metabolic behaviors. *Analytical chemistry*, 82(23):9789–9796, 2010.

[84] Hidenobu Miyazawa and Alexander Aulehla. Revisiting the role of metabolism during development. *Development*, 145(19):dev131110, 2018.

[85] Mourad Mourafiq. Polyaxon: Cloud native machine learning automation platform. Web page, 2017.

[86] Tatwadarshi P Nagarhalli, Vinod Vaze, and NK Rana. Impact of machine learning in natural language processing: A review. In *2021 third international conference on intelligent communication technologies and virtual mobile networks (ICICV)*, pages 1529–1534. IEEE, 2021.

[87] Janina Oetjen, Michaela Aichler, Dennis Trede, Jan Strehlow, Judith Berger, Stefan Heldmann, Michael Becker, Michael Gottschalk, Jan Hendrik Kobarg, Stefan Wirtz, et al. Mri-compatible pipeline for three-dimensional maldi imaging mass spectrometry using paxgene fixation. *Journal of proteomics*, 90:52–60, 2013.

[88] Janina Oetjen, Kirill Veselkov, Jeramie Watrous, James S McKenzie, Michael Becker, Lena Hauberg-Lotte, Jan Hendrik Kobarg, Nicole Strittmatter, Anna K Mróz, Franziska Hoffmann, et al. Benchmark datasets for 3d maldi-and desi-imaging mass spectrometry. *GigaScience*, 4(1):s13742–015, 2015.

[89] Aytuğ Onan and Mansur Alp Toçoğlu. Weighted word embeddings and clustering-based identification of question topics in mooc discussion forum posts. *Computer Applications in Engineering Education*, 29(4):675–689, 2021.

[90] Andrew Palmer, Prasad Phapale, Ilya Chernyavsky, Regis Lavigne, Dominik Fay, Artem Tarasov, Vitaly Kovalev, Jens Fuchser, Sergey Nikolenko, Charles Pineau, et al. Fdr-controlled metabolite annotation for high-resolution imaging mass spectrometry. *Nature methods*, 14(1):57–60, 2017.

[91] Ashish Kumar Patnaik, Prasanta Kumar Bhuyan, and KV Krishna Rao. Divisive analysis (diana) of hierarchical clustering and gps data for level of service criteria of urban streets. *Alexandria Engineering Journal*, 55(1):407–418, 2016.

[92] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.

[93] Nicola Pezzotti, Thomas Höllt, B Lelieveldt, Elmar Eisemann, and Anna Vilanova. Hierarchical stochastic neighbor embedding. In *Computer Graphics Forum*, volume 35, pages 21–30. Wiley Online Library, 2016.

[94] Monika Pietrowska, Hanna C Diehl, Grzegorz Mrukwa, Magdalena Kalinowska-Herok, Marta Gawin, Mykola Chekan, Julian Elm, Grzegorz Drazek, Anna Krawczyk, Dariusz Lange, et al. Molecular profiles of thyroid cancer subtypes: Classification based on features of tissue revealed by mass spectrometry imaging. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, 1865(7):837–845, 2017.

[95] Andrzej Polanski, Michal Marczyk, Monika Pietrowska, Piotr Widlak, and Joanna Polanska. Signal partitioning algorithm for highly efficient gaussian mixture modeling in mass spectrometry. *PloS one*, 10(7), 2015.

[96] Andrzej Polanski, Michal Marczyk, Monika Pietrowska, Piotr Widlak, and Joanna Polanska. Initializing the em algorithm for univariate gaussian, multi-component, heteroscedastic mixture models by dynamic programming partitions. *International Journal of Computational Methods*, 15(03):1850012, 2018.

[97] EO Postma, HJ van den Herik, and LJ van der Maaten. Dimensionality reduction: a comparative review. *Journal of Machine Learning Research*, 10(1–41):66–71, 2009.

[98] Alan M Race and Josephine Bunch. Optimisation of colour schemes to accurately display mass spectrometry imaging data based on human colour perception. *Analytical and bioanalytical chemistry*, 407(8):2047–2054, 2015.

[99] Alan M Race, Andrew D Palmer, Alex Dexter, Rory T Steven, Iain B Styles, and Josephine Bunch. Spectralanalysis: software for the masses. *Analytical chemistry*, 88(19):9451–9458, 2016.

[100] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[101] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[102] Mayra Z Rodriguez, Cesar H Comin, Dalcimar Casanova, Odemir M Bruno, Diego R Amancio, Luciano da F Costa, and Francisco A Rodrigues. Clustering algorithms: A comparative approach. *PloS one*, 14(1):e0210236, 2019.

[103] Giorgio Roffo, Simone Melzi, Umberto Castellani, Alessandro Vinciarelli, and Marco Cristani. Infinite feature selection: a graph-based feature filtering approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4396–4410, 2020.

[104] Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005.

[105] 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.

[106] Sanaiya Sarkari, Chanchala D Kaddi, Rachel V Bennett, Facundo M Fernández, and May D Wang. Comparison of clustering pipelines for the analysis of mass spectrometry imaging data. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4771–4774. IEEE, 2014.

[107] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE, 2011.

[108] Sergio M Savaresi, Daniel L Boley, Sergio Bittanti, and Giovanna Gazzaniga. Cluster selection in divisive clustering algorithms. In *Proceedings of the 2002 SIAM International Conference on Data Mining*, pages 299–314. SIAM, 2002.

[109] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[110] Khawla Seddiki, Philippe Saudemont, Frédéric Precioso, Nina Ogrinc, Maxence Wisztorski, Michel Salzet, Isabelle Fournier, and Arnaud Droit. Cumulative learning enables convolutional neural network representations for small mass spectrometry data classification. *Nature communications*, 11(1):1–11, 2020.

[111] Gil Sharon, Neha Garg, Justine Debelius, Rob Knight, Pieter C Dorrestein, and Sarkis K Mazmanian. Specialized metabolites from the microbiome in health and disease. *Cell metabolism*, 20(5):719–730, 2014.

[112] Aditya Divyakant Shrivastava, Neil Swainston, Soumitra Samanta, Ivayla Roberts, Marina Wright Muelas, and Douglas B Kell. Massgenie: A transformer-based deep learning method for identifying small molecules from their mass spectra. *Biomolecules*, 11(12):1793, 2021.

[113] Rebecca L Siegel, Kimberly D Miller, Hannah E Fuchs, and Ahmedin Jemal. Cancer statistics, 2021. *CA: a cancer journal for clinicians*, 71(1):7–33, 2021.

[114] Tina Smets, Nico Verbeeck, Marc Claesen, Arndt Asperger, Gerard Griffioen, Thomas Tousseyn, Wim Waelput, Etienne Waelkens, and Bart De Moor. Evaluation of distance metrics and spatial autocorrelation in uniform manifold approximation and projection applied to mass spectrometry imaging data. *Analytical chemistry*, 2019.

[115] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[116] Hyuna Sung, Jacques Ferlay, Rebecca L Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray. Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 71(3):209–249, 2021.

[117] Spencer A Thomas, Alan M Race, Rory T Steven, Ian S Gilmore, and Josephine Bunch. Dimensionality reduction of mass spectrometry imaging data using autoencoders. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7. IEEE, 2016.

[118] Tian Tian, Ji Wan, Qi Song, and Zhi Wei. Clustering single-cell rna-seq data with a model-based deep learning approach. *Nature Machine Intelligence*, 1(4):191–198, 2019.

[119] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[120] Herbert Tilg, Niv Zmora, Timon E Adolph, and Eran Elinav. The intestinal microbiota fuelling metabolic inflammation. *Nature Reviews Immunology*, 20(1):40–54, 2020.

[121] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR, 2018.

[122] Dennis Trede, Stefan Schiffler, Michael Becker, Stefan Wirtz, Klaus Steinhorst, Jan Strehlow, Michaela Aichler, Jan Hendrik Kobarg, Janina Oetjen, Andrey Dyatlov, et al. Exploring three-dimensional matrix-assisted laser desorption/ionization imaging mass spectrometry data: three-dimensional spatial segmentation of mouse kidney. *Analytical chemistry*, 84(14):6079–6087, 2012.

[123] Laurens Van Der Maaten. Learning a parametric embedding by preserving local structure. In *Artificial intelligence and statistics*, pages 384–391. PMLR, 2009.

[124] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[125] Jannis van Kersbergen, F Ghazvinian Zanjani, Sveta Zinger, Fons van der Sommen, Benjamin Balluff, DR Naomi Vos, Shane R Ellis, RMA Heeran, Marit Lucas, Henk A Marquering, et al. Cancer detection in mass spectrometry imaging data by dilated convolutional neural networks. In *Medical Imaging 2019: Digital Pathology*, volume 10956, pages 94–101. SPIE, 2019.

[126] Roy Varshavsky, Assaf Gottlieb, Michal Linial, and David Horn. Novel unsupervised feature filtering of biological data. *Bioinformatics*, 22(14):e507–e513, 2006.

[127] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[128] Nico 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.

[129] Kirill A Veselkov, Reza Mirnezami, Nicole Strittmatter, Robert D Goldin, James Kinross, Abigail VM Speller, Tigran Abramov, Emrys A Jones, Ara Darzi, Elaine Holmes, et al. Chemo-informatic strategy for imaging mass spectrometry-based hyperspectral profiling of lipid signatures in colorectal cancer. *Proceedings of the National Academy of Sciences*, 111(3):1216–1221, 2014.

[130] DRN Vos, SR Ellis, Benjamin Balluff, and RMA Heeren. Experimental and data analysis considerations for three-dimensional mass spectrometry imaging in biomedical research. *Molecular Imaging and Biology*, pages 1–11, 2020.

[131] Dongfang Wang and Jin Gu. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics*, 16(5):320–331, 2018.

[132] Piotr Widlak, Grzegorz Mrukwa, Magdalena Kalinowska, Monika Pietrowska, Mykola Chekan, Janusz Wierzgon, Marta Gawin, Grzegorz Drazek, and Joanna Polanska. Detection of molecular signatures of oral squamous cell carcinoma and normal epithelium–application of a novel methodology for unsupervised segmentation of imaging mass spectrometry data. *Proteomics*, 16(11-12):1613–1621, 2016.

[133] Chalini D Wijetunge, Isaam Saeed, Berin A Boughton, Jeffrey M Spraggins, Richard M Caprioli, Antony Bacic, Ute Roessner, and Saman K Halgamuge. Exims: an improved data analysis pipeline based on a new

peak picking method for exploring imaging mass spectrometry data. *Bioinformatics*, 31(19):3198–3206, 2015.

[134] Jason WH Wong, Caterina Durante, and Hugh M Cartwright. Application of fast fourier transform cross-correlation for the alignment of large chromatographic and spectral datasets. *Analytical chemistry*, 77(17):5655–5661, 2005.

[135] Karsten Wüllems, Jan Kölling, Hanna Bednarz, Karsten Niehaus, Volkmar H Hans, and Tim W Nattkemper. Detection and visualization of communities in mass spectrometry imaging data. *BMC bioinformatics*, 20(1):1–12, 2019.

[136] Eric P Xing and Richard M Karp. Cliff: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 17(suppl_1):S306–S315, 2001.

[137] William J Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, 1950.

[138] Weichuan Yu, Baolin Wu, Ning Lin, Kathy Stone, Kenneth Williams, and Hongyu Zhao. Detecting and aligning peaks in mass spectrometry data with applications to maldi. *Computational Biology and Chemistry*, 30(1):27–38, 2006.

[139] 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.

[140] Xiaowei Zhao, Feiping Nie, Rong Wang, and Xuelong Li. Robust fuzzy k-means clustering with shrunk patterns learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

# Appendices

# Abbreviations and Symbols

| | |
|---|---|
| BIC | Bayesian Information Criterion |
| DiviK | Divisive Intelligent K-Means |
| GMM | Gaussian Mixture Model |
| MALDI | Matrix-Assisted Laser Desorption/Ionization |
| MSI | Mass Spectrometry Imaging |
| NNMF | Non-Negative Matrix Factorization |
| OSCC | Oral Squamous Cell Carcinoma |
| PAFFT | Peak Alignment using Fast Fourier Transform |
| PCA | Principal Components Analysis |
| ROI | Region of Interest |
| TIC | Total Ion Count |
| ToF | Time-of-Flight |
| UMAP | Uniform Manifold Approximation and Projection |
| VAE | Variational Autoencoder |

# Attached USB Drive Content

A USB Drive is attached to this work, with following content:

- content of the PhD Thesis in `pdf` format,
- working copy of computational packages' source code,
- Jupyter Notebooks with partial summaries of carried out experiments,
- sample OSCC dataset used in this study.

Source code of the methods used here is available online through GitHub platform (`https://github.com/gmrukwa/divik`).

# List of Figures

# List of Tables