Otto-von-Guericke University Magdeburg

Faculty of Computer Science

Master Thesis

Learning from the Eigenvalues of Metaproteomic Abundance Data - Spectral Clustering and Classification

Author:

Rahul Mondal

March 28, 2023

Advisors:

Prof. Dr. rer. nat. habil. Gunter Saake
Department of Computer Science.
Otto-von-Guericke University Magdeburg, Germany

Jun.-Prof. Dr. Robert Heyer
Faculty of Technology
Bielefeld University, Bielefeld, Germany

Dr. David Broneske
German Center for Higher Education Research and Science Studies (DZHW), Hannover, Germany

Mondal, Rahul:
Learning from the Eigenvalues of Metaproteomic Abundance Data - Spectral Clustering and Classification
Master Thesis, Otto-von-Guericke University Magdeburg, 2023.

Abstract

This thesis presents a study on the application of spectral clustering and classification techniques to metaproteomic abundance data. Metaproteomics is the study of the proteins present in an ecosystem or community, and metaproteomic abundance data refers to the measurement of the relative amounts of different proteins present in a sample. The main objective of the thesis is to investigate the use of eigenvectors and eigenvalues from metaproteomic abundance data and use them to implement unsupervised and supervised learning algorithms. The study first demonstrates the use of spectral clustering, a technique that uses graph Laplacian matrix to capture the local structure of data, and then transforms it into a matrix of eigenvectors to identify low-dimensional embeddings and for further clustering, on metaproteomic abundance data. The second part of the thesis focuses on classification, where the eigenvalues are consequently used to reduce features in metaproteomic abundance datasets and then used as input to classification. The results show that spectral clustering can outperform agglomerative clustering by improving cluster separation by over 50%, and in terms of class separation. Furthermore, it was found that no transformation provides the best cluster separation for spectral clustering when used as a dimension reduction technique prior to clustering, whereas, the principal component analysis provides better clustering results for hierarchical clustering. Additionally, the use of eigenvectors prior to classification showed an increase of 2% in accuracy and 3% of Matthew's Correlation Coefficient.

Acknowledgments

I am deeply grateful to my advisor Jun. -Prof. Dr. Robert Heyer for trusting in my abilities and for constantly providing creative freedom throughout the journey of all the projects and publications leading to this master thesis work. It remains my utmost pleasure to have been able to work under his supervision.

I am equally grateful to Dr. David Broneske for providing valuable insights on writing manuscripts and presenting research, during the time he supervised and co-authored my publication. I have learnt a lot from him regarding research which reflects in the output of this manuscript. His calm composure has provided a standard in my own mind.

I feel blessed to have been supervised by Prof. Dr. Gunter Saake. It was one of my most enthralling opportunities to have worked alongside him on research work and later be supervised by someone of his calibre. His immense contribution to his field has motivated my own career prospects.

My deepest gratitude to Daniel Walke for his valuable feedback and suggestions throughout the course of the thesis and writing of this manuscript. He provided me with factual as well as moral support, when it redeemed necessary, at any given point in this research.

Last but not the least, I am grateful to my friends and family, for constantly believing in me and providing me with unbound support through any journey I had come across, with one of the most important being my master's thesis.

Contents

List of Figures ix

List of Tables xiii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Scope . . . 2
- 1.3 Research Questions . . . 3

2 Background 5
- 2.1 Metaproteomics . . . 5
  - 2.1.1 Biomarker discovery for IBD and NAFLD . . . 7
- 2.2 Graph . . . 8
  - 2.2.1 Graph Data Model . . . 8
  - 2.2.2 Graph Laplacian . . . 10
  - 2.2.3 Graph Database Model . . . 11
- 2.3 Data Transformation: Normalization, Eigendecomposition and PCA . . . 12
  - 2.3.1 Normalization . . . 12
  - 2.3.2 Eigendecomposition . . . 13
  - 2.3.3 Principal Component Analysis (PCA) . . . 13
- 2.4 Unsupervised learning: Clustering . . . 14
  - 2.4.1 Spectral Clustering Algorithms: NJW and Self-Tuning . . . 17
  - 2.4.2 Clustering Evaluation: Internal and External Indices . . . 19
- 2.5 Supervised Learning: Classification . . . 22
  - 2.5.1 Classification Algorithms: Nearest Centroid, k-Nearest Neighbour and Decision Tree . . . 22
  - 2.5.2 Classification Evaluation . . . 26

3 Related Work 27
- 3.1 Clustering of Omics Data . . . 28
  - 3.1.1 Metaproteomics of the gut microbiome of patients with IBD . . . 28
  - 3.1.2 Metaproteomic of the gut microbiome of patients with NAFLD . 29
- 3.2 Spectral Clustering on Omics Data . . . 29
  - 3.2.1 SpectralTAD . . . 29
  - 3.2.2 Spectrum . . . 30

4 Methodology 31
- 4.1 Data Selection and Pre-processing . . . 31

4.1.1 Dataset 1: Metaproteomics of faecal samples of patients with Inflammatory Bowel Diseases. 33
4.1.2 Dataset 2: Discovering Biomarkers for Non-Alcoholic Steatohepatitis Patients using Fecal Metaproteomics 33
4.1.3 Dataset 3: Combining Dataset 1 and Dataset 2 34
4.2 Data Transformation: Normalization, Eigendecomposition and PCA . . . 35
4.3 Experiment 1: Comparison of Clustering 35
4.4 Experiment 2: Comparison of Classification 36
4.5 Hardware and Software 38

5 Evaluation 41
5.1 Experiment 1: Comparison of Clustering 41
5.2 Experiment 2: Comparison of Classification 45
5.3 Comparison of datasets 49
5.4 Discussion of Research Questions 50

6 Conclusion 53
6.1 Scope of Future Work 54

Bibliography 57

List of Figures

1.1 **Classification of meta-omics.** Proteomics and metaproteomics are derived from meta-omics under the context of analysing proteins and their derivatives [Hardouin et al., 2021]. 1

2.1 **General workflow for metaproteomic analysis using mass spectrometry** [Gil, 2017] 6

2.2 **Example of a full-connected, weighted, undirected and homogeneous graph containing patient nodes, with metaprotein abundances and labels as node attributes.** The edge weights are Euclidean distances for each pair of nodes. 8

2.3 **Relational and graphical representation of metaproteomic abundance data.** The columns in the relational table are represented as node attributes (not shown) in the graph. The edge weights represent the similarity between the nodes, often generated by calculating the distance between rows based on column values. 8

2.4 **a) Distance matrix** represents distance/dissimilarity values, with 0 representing maximum similarity **b) Similarity (Affinity) matrix** represents similarity values, with 1 representing maximum similarity. **c)Adjacency matrix** represents adjacency between two nodes and is mostly used in the context of a graph. 9

2.5 **Types of similarity graphs as discussed by von Luxburg** [von Luxburg, 2007] (Figure is borrowed.) 10

2.6 **Evolution of database models.** Solid arrows denote the influence of graph theory while dotted arrows represent the influence of one database model on another [Angles and Gutierrez, 2008]. 11

2.7 **The roadmap to spectral clustering through clustering taxonomy** [Ezugwu et al., 2021] [Aggarwal and Wang, 2010]. 16

2.8 **Step-wise comparison of spectral against basic clustering algorithms.** 17

2.9 **Taxonomy of classification algorithms** [Nicolas, 2015]. 22

2.10 **Components of a decision tree**. The depth is 3 in this case. 24

2.11 **Decision boundaries after training on the iris dataset for** A)Nearest Centroid B) k-Nearest Neighbour C) Decision Tree (Borrowed figures) 25

4.1 **Methodology for this research.** Arrows represent the data flow. Blue boxes denote events used for the clustering tasks and green boxes represent events used for classification tasks only. MCC = Matthew's Correlation Coefficient. 32

4.2 **Flowchart to elaborate the formation of dataset 3.** Arrows represent the data flow. 34

5.1 **Bar plots comparing A) silhouette coefficient and B) adjusted rand index for Dataset 1 subdivided into individual data transformations.** Red dashed lines represent the benchmark silhouette coefficient (0.17) and adjusted rand index (0.31). SC = Spectral Clustering, NJW = Ng-Jordan-Weiss algorithm and ST = Self-Tuning algorithm. 42

5.2 **Bar plots comparing A) silhouette coefficient and B) adjusted rand index for Dataset 2 subdivided into individual data transformations.** Red dashed lines represent the benchmark silhouette coefficient (0.02) and adjusted rand index (0.44). SC = Spectral Clustering, NJW = Ng-Jordan-Weiss algorithm and ST = Self-Tuning algorithm. 43

5.3 **Bar plots comparing A) silhouette coefficient and B) adjusted rand index for Dataset 3 subdivided into individual data transformations.** No benchmark values are available. SC = Spectral Clustering, NJW = Ng-Jordan-Weiss algorithm and ST = Self-Tuning algorithm. . . . 43

5.4 **Box plots comparing the accuracy and MCC for** datasets(A,D), data transformations (B,E) and classification algorithms (C,F). 44

5.5 **Bar plots comparing A) accuracy and B) Matthew's correlation coefficient (mcc) for Dataset 1 subdivided into individual data transformations.** No benchmark values are available. nc = nearest centroid, knn = k-nearest neighbours and dt = decision tree algorithms. 46

5.6 **Bar plots comparing A) accuracy and B) Matthew's correlation coefficient (mcc) for Dataset 2 subdivided into individual data transformations.** Benchmark values for dataset 2 are compared in figure 5.7. nc = nearest centroid, knn = k-nearest neighbours and dt = decision tree algorithms. 46

5.7 **Bar plots comparing A) accuracy and B) Matthew's correlation coefficient (mcc) for Dataset 2 subdivided into individual data transformations.** Red dashed lines represent the benchmark accuracy (0.86). No benchmark mcc is available. nc = nearest centroid, knn = k-nearest neighbours and dt = decision tree algorithms. 47

5.8 **Bar plots comparing A) accuracy and B) Matthew's correlation coefficient (mcc) for Dataset 3 subdivided into individual data transformations.** Red dashed line represents the benchmark accuracy (0.86). No benchmark mcc is available. nc = nearest centroid, knn = k-nearest neighbours and dt = decision tree algorithms.. 47

5.9 **Box plots comparing the accuracy and MCC for** datasets(A,D), data transformations (B,E) and classification algorithms (C,F). 48

5.10 **Box plots comparing the A) precision and B) recall**. for the 3 datasets. 49

List of Tables

2.1 **Graph vs Relational Database** [Vicknair et al., 2010] [Khan et al., 2019] . . . 12

2.2 **Comparison of normalization and PCA.** . . . 13

2.3 **Comparison of clustering algorithms** Garima et al. [2015] Celebi [2014] n = number of data points, N = number of links, C = Number of link clusters, T = Number of iterations . . . 16

2.4 **Comparison of spectral clustering algorithms** [Nascimento and De Carvalho, 2011] . . . 19

2.5 **Clustering validation indices** [Gan et al., 2007] [Brun et al., 2007]. . . 21

2.6 **Comparison of classifying algorithms** [Alpaydin, 2020]. . . . 23

2.7 **Confusion matrix for comparing predicted and actual labels.** . . . 26

3.1 **List of related work discussed in this section.** . . . 27

4.1 **Comparison of datasets used in this research.** IBS = Inflammatory Bowel Diseases, NAFLD = Non-Alcoholic Fatty Liver Diseases. . . . 32

4.2 **Overview of python packages and their versions used in the experiments.** . . . 39

5.1 **Mean silhouette and adjusted rand coefficients for all categories: datasets, data transformations and clustering algorithms.** . . . 44

5.2 **Mean accuracy and Matthew's correlation coefficient (mcc) for all categories: datasets, data transformations and clustering algorithms** 48

1. Introduction

1.1 Motivation

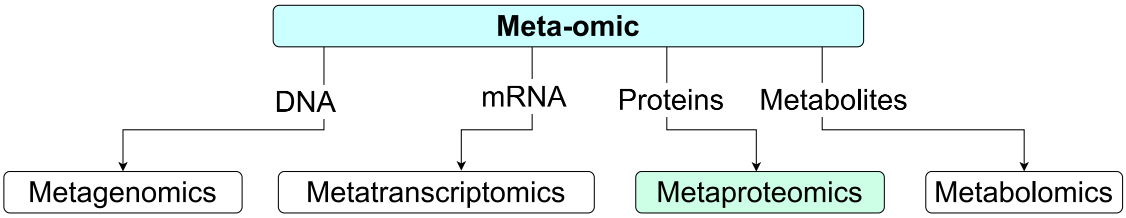


Figure 1.1: Classification of meta-omics. Proteomics and metaproteomics are derived from meta-omics under the context of analysing proteins and their derivatives [Hardouin et al., 2021].

The study of the total amount of genetic material (metagenomics), protein content (metaproteomics), and metabolites (metabolomics) contained in a complex biological sample is referred to as meta-omics (see Figure 1.1). It is a thorough method of comprehending the makeup and usefulness of microbial communities, such as those found in soil, water, the human gut, or other environments. The study of all the genetic material (DNA) in a microbial population, whether it comes from a recognized species or not, is known as metagenomics. This method can shed light on the diversity, organization, and possible utility of microbial communities. It offers details on the produced proteins and the tasks that they are carrying out. The study of metatranscriptomics entails the examination of the entire collection of RNA transcripts (transcriptomes) found in complicated biological samples like microbial communities or environmental samples. This method offers details on the functional actions and gene expression patterns of the organisms in the sample. The analysis of all the small molecules (metabolites) produced by a microbial population is known as metabolomics. It may reveal details about the biological processes and metabolic processes taking place in the neighbourhood.

The abundance of proteins in a sample can provide valuable insights into samples taken from patients. However, analyzing this data can become challenging due to the vast amount of metaproteins that are present in each sample being measured. This project aims to investigate the use of eigenvalues in the group and classify patients based on eigenvalues of metaproteomic abundance data.

Data in metaproteomics are highly interlinked, i.e. several metaproteins commonly abundant for a set of patients and a massive set of metaproteins that could possibly be found in the human body (>10,000 [1] different types of proteins). However, the choice of a data structure and/or a database system is often limited to two options - relational, graph, key-value store, document store etc. Traditionally, relational data structures are highly scalable and relatively easy to analyse. However, this comes at the cost of low flexibility since relational data structures can only rely on foreign keys to associate two tables. Graph data structures help to overcome this using the node-edge schema (see Figure 2.3). This helps to upload data with a flexible schema in the form of node attributes, edge weights etc as well as having the possibility to form directed graphs. Additionally, the node-edge structure provides vivid data visualisation, making it easier to interpret the results.

However, the challenge of analysing highly interlinked and high-dimensional (high number of columns) metaproteomic abundance data is not completely overcome by the flexible data schema provided by graphical data structures. It requires a detailed analysis with specialized algorithms, e.g. dimension reduction techniques, and clustering algorithms, that can discover hidden similarities and differences within metaproteomic abundance data. Therefore, I have investigated a popular clustering algorithm for graph data structure, spectral clustering.

Spectral clustering is a clustering technique that can group similar data points/ nodes/ rows together into clusters by constructing a similarity matrix of a graph, transforming it into a matrix of eigenvectors, and using these eigenvectors as input to a clustering algorithm.In this thesis, I have applied spectral clustering on metaproteomic abundance data to cluster similar patients with a higher cluster separation achieved than hierarchical clustering. Additionally, I exploited the eigendecomposition in spectral clustering for classification and improved the prediction accuracy of classifying different patients. To summarise, I have demonstrated the potential of eigenvalues to extract relevant information from metaproteomic abundance data and improve the grouping of patients using clustering algorithms, and further classify them using machine learning algorithms. This is beneficial in providing a comprehensive overview of the possible methods and best practices one might come across while investigating metaproteomic data.

1.2 Research Scope

In this study, using metaproteomic abundance data from healthy and diseased individuals that are suffering from inflammatory bowel diseases (IBD) and non-alcoholic steatohepatitis (NASH), I contribute the following:

1. A detailed analysis of the implementation of two spectral clustering algorithms, Ng-Jordan-Weiss (NJW) and Self-tuning (ST), on labelled metaproteomic abundances to group control and diseased patients, with a focus on:

[1] Retrieved from https://www.mpg.de/11447687/W003_Biology_medicine_054-059.pdf

   (a) Data pre-processing and transformations: original, normalization and principal component analysis (PCA).
   (b) Number of used eigenvectors: k and 2k, where k is the number of desired clusters

2. A detailed analysis of a classification pipeline, where eigendecompositions (generation of eigenvalues and eigenvectors from the dataset) from spectral clustering algorithms were used as data transformation for metaproteomic abundances, to predict control and diseased patients. The following classifiers were used for this investigation:

   (a) Nearest Centroid Classifier (NC).
   (b) k-nearest neighbour classifier (k-NN).
   (c) Decision Tree (DT).

3. A method to combine metaproteomic abundance data from two separate use cases i.e., separate datasets observing different sets of diseases, through the intersection of common metaproteins (columns), to improve prediction quality by achieving higher accuracy and Matthews Correlation Coefficient as compared to the original datasets.

1.3 Research Questions

Within the scope of the research, I have answered the following research questions in this thesis:

Research Question 1: How well does spectral clustering group patients from metaproteomic abundances, in terms of internal and external validation indices, in comparison to hierarchical clustering?

To answer the first research question, I applied hierarchical and spectral clustering on metaproteomic abundances and evaluated the resulting partitions, using internal and external clustering evaluation indices. These helped to compare the chosen algorithms, firstly on how well the resulting clusters or partitions are separated without (internal indices) and with (external indices) the class labels being considered. While generally, a high score of internal indices is proof of good partitioning, any improvement in the separation of class labels by a clustering algorithm would also indicate that the algorithm can capture the underlying structure as formed due to the class labels and would be efficient pre-cursors to supervised learning.

Research Question 2: To what extent do data transformation techniques such as normalization and PCA improve clustering performance for metaproteomic abundances, in terms of internal and external validation indices?

To answer research question 2 I investigated several kinds of data transformations whose goal is to reduce biases and computational expenses for running complex algorithms. To find an optimal direction for finding the right type of data transformation for metaproteomic abundance data, I applied normalization and PCA to compare the performance of normalization and dimension reduction on the data. Most often it is quite useful to normalize the data to bring out hidden correlations and detect outliers more efficiently. However, it is also interesting to know whether further reducing the dimensions through PCA would contribute to improving cluster separation and prediction quality.

Research Question 3: How much better accuracy and Matthews Correlation Coefficient could be achieved over normalization, if eigendecomposition was applied as a data transformation step, to predict patient labels from metaproteomic abundances?

After having satisfactory cluster separation as well as class label separation through spectral clustering, I used the eigendecomposition used in the self-tuning spectral clustering algorithm as a data pre-processing step. Then I applied classic machine learning algorithms (nearest centroid, k-nearest neighbour and decision tree algorithm) to realize the potential of the eigendecomposition of metaproteomic abundances, and to improve our quantitative and predictive analyses of such data.

Research Question 4: How much improvement could be achieved, in terms of clustering and classification, if two metaproteomic abundances datasets were combined into one?

And finally, I investigated whether it is an efficient method to combine metaproteomic abundance data from two separate use cases into one and implement classification algorithms. To verify this, I have measured the silhouette coefficient, adjusted rand index, accuracy and MCC values for all 9 labels in the combined dataset 3. However, it could be possible that any improvement can only be noticed for the control label, which was present for both datasets 1 and 2. As a result, I have also measured the precision and recall for only the control label, after performing classification on dataset 3.

In the following chapter 2, I have provided background on the following topics:

1. Metaproteomics.
2. Graph Data and Database, and Graph Laplacian.
3. Data Transformation: Normalization, Eigendecomposition and Principal Component Analysis (PCA).
4. Unsupervised Learning: Spectral Clustering.
5. Supervised Learning: Nearest Centroid, k-Nearest Neighbour and Decision Tree Classifiers.

Then in chapter 3, I have discussed similar research that has helped shape my own methodology, which I have elaborated in chapter 4. Next in chapter 5, I have presented and discussed the results of my research as well as concluded with the scope for future research.

2. Background

In this section, I have provided a detailed overview of the following topics: metaproteomics, graph, clustering and classification, to help understand the key aspects of the methodology for this research.

2.1 Metaproteomics

Proteins are one of the primary building blocks of all life on earth and play a crucial role in several biological functions and processes. Proteins can be broken down into amino acids and peptides serve as an intermediate component and a primary derivative in the process of extracting amino acids from proteins. Changes in the abundance levels, structure, or function of proteins can indicate the presence of a disease. Proteomics is the study of the complete set of proteins, called the proteome, that is expressed by a cell, tissue, organism, or biological sample.

Large-scale investigations of biological molecules—such as genes, proteins, metabolites, and other biomolecules—and their interactions with one another are referred to as ”omic” research. The use of high-throughput technologies to analyze and interpret massive amounts of data in order to provide researchers with a thorough knowledge of biological systems is referred to as an ”omics” study. Genomic, transcriptomic, proteomic and metabolomic investigations are a few examples of omics research. Each of these omics strategies concentrates on a distinct class of biological molecules and offers a unique viewpoint on the biological system under investigation.

Metaproteomics focuses on the study of all proteins present in a complex mixture of biological samples, e.g., soil, water, and gut contents. It helps to understand the functional roles and interactions of expressed proteins within a complex community, providing insights into the metabolic and ecological processes that occur within a given ecosystem. Metaproteomics differs from traditional proteomics in that it focuses on the total protein complement of a sample, rather than the study of a single organism or defined set of proteins. The data generated by metaproteomics can provide valuable information for fields such as environmental science, microbiology, and biotechnology. In metaproteomics, primary derivatives are the proteins that are identified directly from the mass spectrometry data or other techniques used to analyze the metaproteins, which are groups of functionally similar proteins. These proteins are typically identified through a process of peptide sequencing, in which the peptides generated by the digestion of the metaproteins are identified and assembled into complete protein sequences. A general workflow of metaproteomic protein identification using mass spectrometry is shown in figure 2.1.

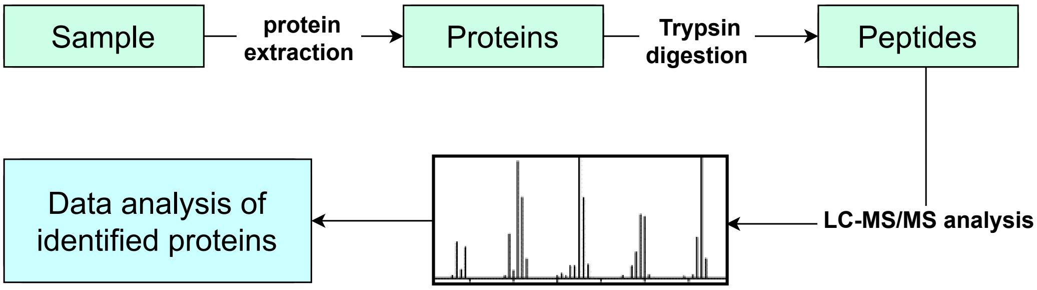


Figure 2.1: General workflow for metaproteomic analysis using mass spectrometry [Gil, 2017]

Once the primary derived proteins are identified, researchers may perform further analyses, such as functional annotation or network analysis, to gain insights into the metabolic activities and interactions of the microorganisms in the community. Additionally, researchers may perform validation studies to confirm the identity and function of the primary derived proteins, which may involve additional experiments such as Western blotting or enzyme assays. The data collection for metaproteomics involves the following steps as discussed by Ngom et al. [2019] (see figure 2.1):

1. **Sample collection** from the soil, water, human gut etc.
2. **DNA extraction** from sample to determine microbial community composition.
3. **Proteins extraction and purification** through methods such as boiling, detergent treatment or mechanical lysis.
4. **Peptide extraction** from proteins using enzymes.
5. **Separation of peptides** using methods such as chromatography prior to techniques such as 2D gel electrophoresis, mass spectrometry (MS) or liquid chromatography-MS (LC-MS).
6. **Protein identification** through comparison of sequences in a database.

Metaproteomic analysis has been potentially useful to identify biomarkers of diseases. Additionally, metaproteomics can also aid in identifying proteins that are essential for the survival of specific microbial species. This information can be used to develop targeted therapies that disrupt the function of these proteins and inhibit the growth or activity of specific microorganisms. For example, gut microbiome analysis has been used to identify patients with disease as well as potential drug targets in the gut microbiome for the treatment of inflammatory bowel disease or other gastrointestinal disorders [Ngom et al., 2019].

While dealing with metaproteomic data, there could be two terms potentially acting as a major source of confusion for non-domain specialists: spectrum from spectrometry and spectral from spectral clustering. Spectrum refers to the collection of values representing a signal or data point as a function of frequency, and especially in the context of mass spectrometry, the term refers to the collection of peaks that represent the abundance of different peptides in the sample being examined. Spectral on the other hand refers to being related to the spectrum, and in the context of spectral clustering, it refers to the algorithm operating on the similarity matrix derived from spectral decomposition. Examples of work done in the context of mass spectrometry spectrum include MetaLab [Cheng et al., 2017] and msCRUSH [Wang et al., 2018]. MetaLab is a pipeline for protein identification, quantification and taxonomic profiling based on peaks observed in mass spectrometry data. msCRUSH is a software, compiled in C++, where a locality-sensitive hashing technique is used for clustering spectrum and consequently, peptide identification. Such work is not to be confused with spectral clustering on abundance data (related work discussed in section 3), where clustering is applied after the proteins/peptides have been identified for grouping similar proteins, and not during the identification process.

2.1.1 Biomarker discovery for IBD and NAFLD

Inflammatory bowel disease (IBD) [Zhang and Li, 2014] and nonalcoholic fatty liver disease (NAFLD) [Neuschwander-Tetri, 2017] are two chronic inflammatory conditions that affect the gastrointestinal tract and liver, respectively. IBD is an umbrella term and contains diseases under the heading of e.g., Crohn's Disease (CD) and Ulcerative Colitis (UC). In the research by Lehmann et al. [2019], two forms of Ulcerative Colitis patients were observed - ulcerative colitis in the active stage (UCa) and remission stage (UCr). Patients with IBD also have an increased risk of developing benign and/or malignant tumours such as Colon Adenoma (CA) and/or Gastric Carcinoma (GCA), respectively [Lehmann et al., 2019]. Metaproteomic analysis of stool samples from IBD patients has identified a number of potential biomarkers, including proteins involved in inflammation, immune response, and intestinal barrier function. These biomarkers have the potential to aid in the diagnosis, prognosis, and monitoring of IBD, as well as in the development of new therapeutics. [Lehmann et al., 2019] also observed whether a chronic condition known as Irritable Bowel Syndrome (IBS), characterized by recurrent abdominal pain, could be distinguished through metaproteomic of faecal samples from patients with the above diseases.

NAFLD, similarly consists of diseases like nonalcoholic steatohepatitis (NASH) and hepatocellular carcinoma (HCC). Non-alcoholic steatohepatitis (NASH) is a chronic liver disease that is associated with obesity, insulin resistance, and metabolic syndrome. It may develop into cirrhosis, severe fibrosis, or even hepatocellular carcinoma. (HCC). To increase survival rates for NASH patients, early detection of HCC is essential, but existing diagnostic techniques are insufficient. Metaproteomic analysis can be used as a powerful tool for biomarker discovery in these diseases, as it allows for the identification and quantification of the complete set of proteins expressed by the microorganisms in the gut or liver. Metaproteomic analysis of liver biopsies, serum and/or faecal samples from NAFLD patients have identified a range of potential biomarkers, including proteins involved in lipid metabolism, oxidative stress, and inflammation. These biomarkers may aid in the diagnosis and monitoring of NASH and HCC, as well as in the development of new therapies.

2.2 Graph

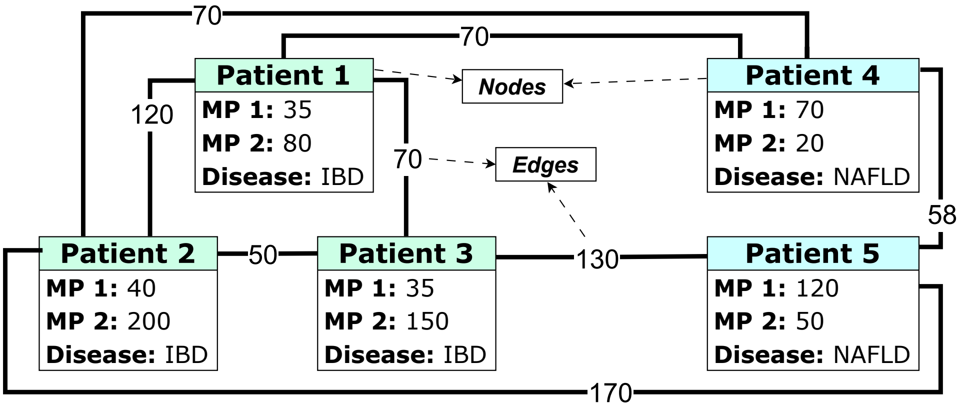
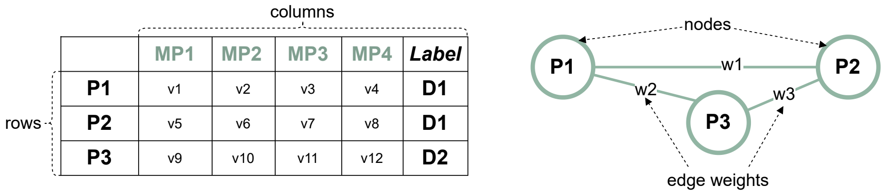


Figure 2.2: Example of a full-connected, weighted, undirected and homogeneous graph containing patient nodes, with metaprotein abundances and labels as node attributes. The edge weights are Euclidean distances for each pair of nodes.

2.2.1 Graph Data Model

A graph (figure 2.2), often represented by, G = (V,E), where V denote vertices (nodes/data points) and E define edges (relationships). It is based on graph theory in discrete mathematics and has proven to be useful in the domains of database design, software engineering, circuit designing, network designing and visual interfaces. In addition, it has inspired the conceptualization of several database models e.g., graph database, XML as well as data structures e.g., trees and linked lists. Graph nodes can contain node attributes comparable to column/feature values in relational tables.

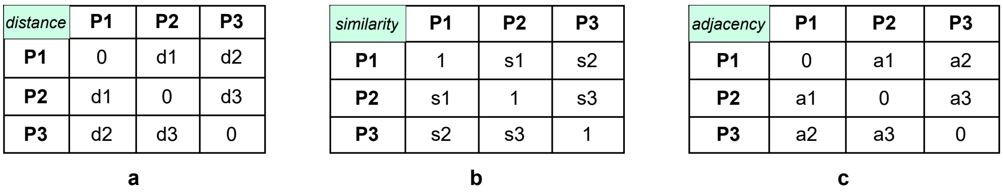
A graph can be undirected, as well as directed meaning the edges contain directional relationships. However, for spectral clustering, edges should be undirected, since the similarity matrix required for clustering should be symmetric which is not the case for directed graphs. Additionally, edges in graphs for spectral clustering contain edge weights that denote the similarity between the connected nodes. Relational tables can be represented as graph nodes as depicted in the figure 2.3 below:



	MP1	MP2	MP3	MP4	*Label*
P1	v1	v2	v3	v4	D1
P2	v5	v6	v7	v8	D1
P3	v9	v10	v11	v12	D2

Figure 2.3: Relational and graphical representation of metaproteomic abundance data. The columns in the relational table are represented as node attributes (not shown) in the graph. The edge weights represent the similarity between the nodes, often generated by calculating the distance between rows based on column values.

To create a graph one would first require to calculate the similarity, adjacency or distance between each pair of rows/samples/measurements and construct an adjacency matrix. An adjacency matrix is most often used in the context of nodes, especially graph nodes. If two nodes are connected, the adjacency is 1, otherwise 0. However, the diagonal of the adjacency matrix contains a piece of different information, which is whether the corresponding node has a self-loop. A self-loop is only present in directed graphs, and hence, for undirected graphs, the diagonal value would always be 0 in an adjacency matrix.



distance	P1	P2	P3
P1	0	d1	d2
P2	d1	0	d3
P3	d2	d3	0

a

similarity	P1	P2	P3
P1	1	s1	s2
P2	s1	1	s3
P3	s2	s3	1

b

adjacency	P1	P2	P3
P1	0	a1	a2
P2	a1	0	a3
P3	a2	a3	0

c

Figure 2.4: a) Distance matrix represents distance/dissimilarity values, with 0 representing maximum similarity **b) Similarity (Affinity) matrix** represents similarity values, with 1 representing maximum similarity. **c)Adjacency matrix** represents adjacency between two nodes and is mostly used in the context of a graph.

Similarity measures such as cosine similarity and the Jaccard coefficient are also used to create graphs. Higher the proximity, the lesser the value of similarity between two nodes. Note that there is a difference between forming a distance, similarity/affinity and adjacency matrix as shown in figure 2.4. However, distance measures such as euclidean distance, manhattan distance can be manipulated to calculate the similarity between two nodes/rows/samples, as the similarity is the inverse of the distance.

$$Edist_{(\mathrm{a,b})} = ||a - b|| \tag{2.1}$$

where Edist = Euclidean distance.

Ideally the diagonal of a similarity matrix contains the maximum similarity between two nodes, which is 1. In the context of spectral clustering and graph nodes, however, the diagonal of a similarity matrix contains the degree of a node, which is the number of edges connected to the corresponding node. The similarity values between two nodes are used to create the similarity graph for all the nodes in a given dataset. There could be three types of similarity graphs that could be created from tabular data as discussed by von Luxburg [2007] (see figure 2.5):

1. **ϵ-neighbourhood graphs** are generated by connecting data points with similarity above a certain threshold ϵ.
2. **k-neighbourhood graphs** are generated using the k-nearest neighbour algorithm. It can be either unweighted or weighted.

    - **Mutual k-nn graphs** are an improvement over k-neighbourhood, where, for any pair of samples, the k-neighbours criteria is checked for both points, should an edge be formed amongst them.

3. **Fully connected (complete) graphs** are generated using similarity functions to calculate similarity (edge weights) between each pair of nodes. In a fully connected graph, there are edges between every pair of nodes present in the graph.

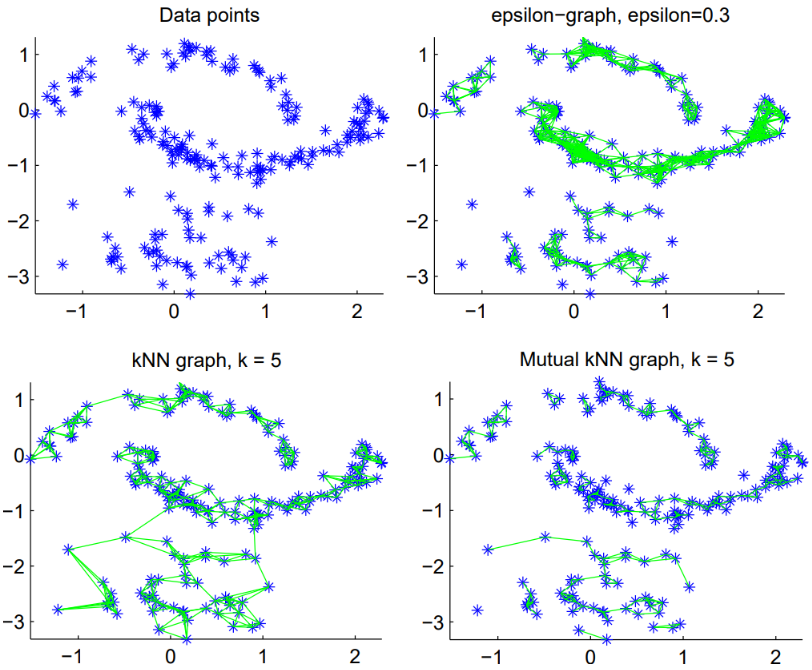


Figure 2.5: Types of similarity graphs as discussed by von Luxburg [von Luxburg, 2007] (Figure is borrowed.)

In the context of this thesis, two types of graphs were used: k-neighbourhood graphs from the default implementation of the python package, and fully-connected graphs as generated by the NJW and Self-tuning spectral clustering algorithms. While fully-connected graphs have a higher space complexity, the larger amount of stored information for each pair of nodes also provides better scope to analyse more accurately. However, for datasets with a high number of columns, this also increases the computational expensiveness in terms of time.

2.2.2 Graph Laplacian

Graph laplacian is a mathematical tool, mostly used in spectral graph theory, represented as a matrix which encodes relationships between nodes in a graph. It can be derived from the adjacency matrix of a graph, as well as the similarity (affinity) matrix. There are several types of Laplacian matrix that can be generated from a given graph [von Luxburg, 2007] [Filippone et al., 2008]:

- Unnormalized Graph Laplacian Matrix L
- Normalized Graph Laplacian Matrices:
  - Symmetric, $L_{sym} = D^{-1/2}LD^{-1/2}$
  - Random Walk, $L_{rw} = D^{-1}L$
- Generalized Graph Laplacian Matrix, $L_G = D^{-1}L = L_{rw}$
- Relaxed Laplacian, $L_\rho = L - {}_\rho D$

where D = Diagonal Matrix, and A = Adjacency Matrix or Affinity Matrix

One of the most useful properties of a graph laplacian is the ability to derive eigenvectors and eigenvalues from it, which can be further exploited by dimension-reduction techniques or clustering algorithms. In the context of spectral clustering, all of the above graph laplacian could be used.

2.2.3 Graph Database Model

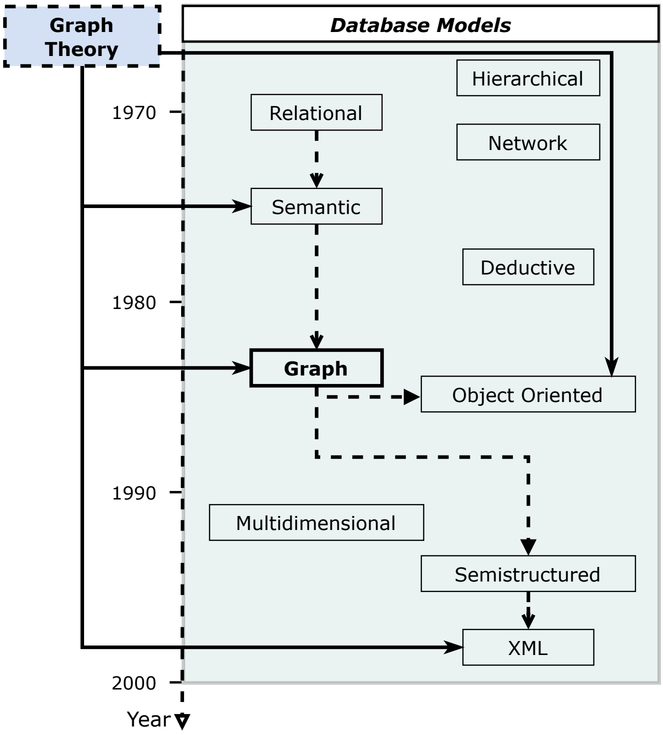


Figure 2.6: Evolution of database models. Solid arrows denote the influence of graph theory while dotted arrows represent the influence of one database model on another [Angles and Gutierrez, 2008].

Graph databases provide an optimized environment to implement and analyse using graph data structures. In addition, to the flexibility of schema, graph databases are horizontally scalable. Horizontal scaling, also known as scaling out, is a way to improve the performance and capacity of a system by distributing the workload across multiple machines rather than relying on a single machine. Horizontal scaling can be achieved in various ways:

- **Sharding** involves dividing the database into smaller subsets or shards, which are distributed across multiple servers.
- **Replication** involves creating copies of the database on multiple servers, which can improve fault tolerance and availability.
- **Partitioning** involves dividing the data into smaller partitions, which can be stored and processed separately by different servers.

.

Graph database models are a product of graph theory (figure 2.6), providing users with the ability to store and query graph data, examples including Neo4j, Tiger Graph etc. One major difference between relational and graph databases is that relational databases comply with ACID which is an acronym for atomicity, consistency, isolation, and durability. On the other hand, most often graph databases comply with BASE which is an acronym for basic availability, soft-state and eventual consistency. However, on some occasions, graph databases can be both ACID and BASE compliant which is the case for Neo4j. A comparative analysis of the graph databases against the relational databases is shown in the table 2.1.

Table 2.1: Graph vs Relational Database [Vicknair et al., 2010] [Khan et al., 2019]

	Relational	**Graph**
Transaction Model	ACID	BASE
Query Language	SQL	Cypher, GQL, SPARQL etc.
Scalability	vertical	horizontal
Integrity Constraints	yes	yes
Flexibility	less mutable schema	easily mutable schema

Additionally, the high flexibility in data schema allows for better representation of highly interlinked data making it a more efficient choice to analyse omic data. There also exists a variety of query languages that are available for various graph databases, with each database allowing for one or a subset of the languages made available. Similar to relational databases, graph databases contain integrity constraints, which ensure data quality and prevent errors, inconsistencies, and data corruption.

2.3 Data Transformation: Normalization, Eigendecomposition and PCA

Data transformations are vital pre-processing techniques in the field of data analysis and machine learning to reduce complexity and capture underlying relationships between features. In the following table 2.2, I have compared three popular data transformation techniques, that I have also used in the experiments of this research.

2.3.1 Normalization

Normalization is a data transformation technique that is used to scale data so that it falls within a certain range or has a certain distribution. It is possible to normalize data using a number of techniques, such as min-max normalization, z-score normalization, and log normalization. To make sure that data from various sources can be compared and analyzed in a meaningful manner, normalization is frequently used. I have used the min-max normalization method for both clustering and classification tasks. This method scales the values of a column between 0 and 1 using the formula in equation 2.2.

$$Scaled\ Value = \frac{Value - Min\ Value}{Max\ Value - Min\ Value} \tag{2.2}$$

Table 2.2: Comparison of normalization and PCA.

	Description	Advantage
Normalization	normalization columns by scaling values between 0 and 1	faster convergence for predictive analytics
Eigendecomposition	lower dimensional representations capturing linear relationships	simplifies complexity of multiple dimensions
Principal Component Analysis	lower dimensional representations capturing the maximum variance	simplifies complexity of multiple dimensions

where Min Value and Max Value refer to the minimum and maximum value in a column. This is useful in ensuring that algorithms treat the features on a similar scale and thus improve performance (faster convergence).

2.3.2 Eigendecomposition

Eigendecomposition is a matrix factorization technique that breaks down a matrix into its constituent parts, namely the eigenvectors and eigenvalues. Eigendecomposition is often used in linear algebra and signal processing to analyze the properties of a matrix, and it can be used to identify the most important features or dimensions in a dataset.

There are two types of eigendecomposition available [Lewis, 2003]:

1. **Symmetric eigendecomposition**: Applied to symmetric matrices, in this case, the eigenvectors are guaranteed to be orthogonal to each other, where the original matrix is decomposed into a set of eigenvectors and a diagonal matrix of eigenvalues.
2. **Non-symmetric eigendecomposition**: Applied to non-symmetric matrices, the eigenvectors may not be orthogonal to each other. However, similar to symmetric, the original matrix is decomposed into a set of eigenvectors and a diagonal matrix of eigenvalues.

In the context of PCA and spectral clustering, symmetric eigendecomposition is used.

2.3.3 Principal Component Analysis (PCA)

PCA is a data transformation technique that uses symmetric eigendecomposition to reduce the dimensionality of a dataset. In PCA, the dataset is projected onto a new set of orthogonal axes, called principal components, that capture the maximum amount of variation in the data. This is achieved by calculating the covariance eigenvectors of the covariance of the input features. This reduces the number of variables in the dataset while retaining the most important information. The steps for PCA are given in the algorithm 2.3.1

Algorithm 2.3.1 Principal Component Analysis (PCA)[Smith, 2002]

```
Input: Dataset containing samples as rows and features as columns.
Center the data by subtracting the mean of each variable from each observation.
Compute the covariance matrix of the centred data.
Compute the eigenvectors and eigenvalues of the covariance matrix.
Sort the eigenvalues in decreasing order and sort the corresponding eigenvectors accordingly.
Choose the number of principal components to retain based on the amount of variance explained or some other criteria.
Project the original data onto the selected principal components.
Output: Projected data on the principal component axes.
```

2.4 Unsupervised learning: Clustering

Clustering is widely used in machine learning and data analysis to partition data points into groups such that points within the same group are more similar to each other than points in different groups. While primarily used in the field of biology to classify species based on their similarities and differences, it was introduced widely in statistics through algorithms such as ISODATA [Ball and Hall, 1965] and Lloyd's algorithm [Lloyd, 1982], both of which were primal versions of the famous k-means algorithm. The k-means algorithm is a basic centroid-based clustering that divides rows into equal partitions using the euclidean distance between each pair of points. Centroid refers to the most average representative point for a set of points or clusters and is calculated by averaging all the feature values for all rows/samples. K-means clustering is used as the final step in spectral clustering algorithms to cluster generated eigenvectors from graphs. The steps of k-means clustering are provided in algorithm 2.4.1.

Algorithm 2.4.1 K-means Algorithm [MacQueen, 1965]

```
1: Input: Dataset containing samples rows and features as columns, and the number of desired clusters k.
2: Initialize k centroids randomly.
3: Calculate the euclidean distance between each point and the initialized centroids.
4: Assign data points to the nearest centroid which are representative of their respective cluster.
5: Recalculate centroids based on the assigned data points to each centroid/cluster.
6: Repeat steps 3, 4 and 5 until the centroids no longer change or a certain number of iterations are completed.
7: Output: k partitions.
```

Applying k-means directly on metaproteomic abundance data would not produce optimal results since k-means work best for normally distributed features while abundance distributions are often skewed. Moreover, k-means is optimal for data whose underlying structure can be efficiently represented by euclidean distance, which is not the case for abundance data. However, eigenvectors can be well separated using k-means since they are low-dimensional and are often fragmented in equal partitions. The ideal clustering workflow ensures an optimal selection of algorithms based on the given datasets and evaluation of them using evaluation metrics. Generally speaking, any cluster analysis method will contain the following steps:

1. **Feature Selection or Extraction:** Extraction distinguishable features (columns) from the original set of variables, which aids in forming and interpreting distinct clusters.
2. **Clustering Algorithm Implementation:** Implementation of algorithms, along with necessary parameters and proximity measure selection such as distance or similarity between two data points, is necessary to decide which algorithm provides the best extraction of characteristics.
3. **Cluster Validation**: While different clustering algorithms may generate different types of clusters for a given dataset, it is important to measure the quality of clusters generated by the used algorithm. Clustering validation measures could be classified into three categories [Brun et al., 2007]: external, internal and relative indices.
4. **Result Interpretation:** Finally the results, with the interpretation by domain experts, could be displayed with meaningful visualisation, for users to interpret.

Clustering algorithms can be broadly generalised into two types: partitional and hierarchical clustering algorithms. Hierarchical clustering such as agglomerative clustering (see algorithm 2.4.2), iteratively groups individual data points in a nested structure, till all points belong to a single cluster. The results of hierarchical clustering can be viewed on a dendrogram. In contrast to hierarchical clustering, partitional clustering divides the data into predetermined groups of clusterse.g., K-means. Hierarchical clustering divides or merges data points recursively, till all points belong to the same cluster (agglomerative) or each point is a singular cluster (divisive). A detailed taxonomy of clustering algorithms is shown in figure 2.7.

Algorithm 2.4.2 Agglomerative Clustering Algorithm [Gower and Ross, 1969]

```
1: Input: Dataset containing samples rows and features as columns, and the number
   of desired clusters k.
2: Initialize by considering each point as a separate cluster.
3: Calculate the pairwise similarity between each pair of clusters using a distance
   metric such as Euclidean distance, cosine similarity etc.
4: Merge the two most similar clusters into a new cluster based on linkage criteria
   such as single, complete, Ward's linkage etc.
5: Repeat 3-4 until only one cluster remains, or define a stopping criterion based
   on the number of desired output clusters k or any measure of clustering quality.
6: Output: k partitions.
```

Hard clustering is a type of partitional clustering, which requires the input of a pre-determined set of clusters, and ensures that every data point belongs to a single cluster only. Its prime competitor, fuzzy clustering, assigns the degree of membership for each point to all possible clusters. K-means is a type of hard clustering, as well as a squared-error minimization technique since it focuses on reducing the sum of

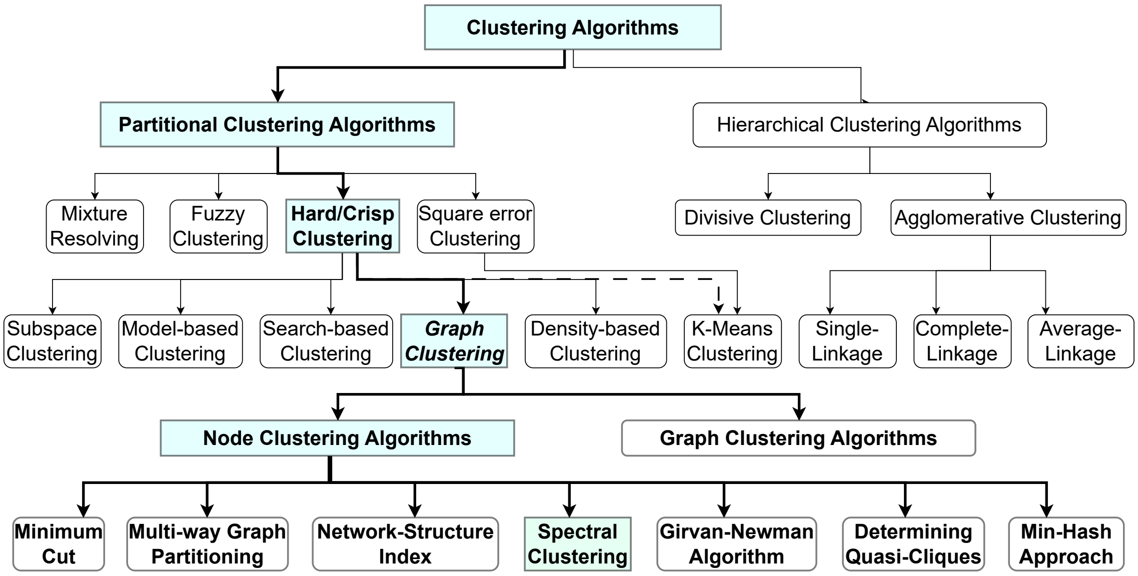


Figure 2.7: The roadmap to spectral clustering through clustering taxonomy [Ezugwu et al., 2021] [Aggarwal and Wang, 2010].

squared errors while assigning every data point a cluster. There also exist density-based clustering techniques such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and OPTICS (Ordering Points To Identify the Clustering Structure) which are good at detecting regions of high-density and low-density in the given data and cluster accordingly. A detailed comparison of spectral clustering algorithm against k-means, OPTICS, fuzzy c-means and single-linkage agglomerative clustering is provided in table 2.3.

Table 2.3: Comparison of clustering algorithms Garima et al. [2015] Celebi [2014]
n = number of data points, N = number of links,
C = Number of link clusters, T = Number of iterations

Type of Clustering			**Parameters**	**Complexity** (Time)	**Pros**	**Cons**
Partitional	**Hard**	**SSE minimization** (k-means)	number of clusters	$\mathbf{O(n^2)}$ [Ahmed et al., 2020]	can efficiently handle large amount of data	cannot handle non-circular clusters
		Density-based (OPTICS)	minimum points to form a cluster	$\mathbf{O(n^3)}$ [Ankerst et al., 1999]	can handle arbitrary shapes of data	computationally expensive
		Graph (spectral)	number of clusters, neighbourhood graph	$\mathbf{O(n^3)}$ [Fujita, 2021]		
	Fuzzy (fuzzy c-means)		fuzzifier (m) membership value (u)	**O(NCT)** [Zhang and Shen, 2018]	data points can belong to multiple clusters	performance depends on initialization
Hierarchical		**Agglomerative** (single-linkage)	linkage criterion	$\mathbf{O(n^3)}$ [Manning et al., 2019]	do not require initialization	computationally expensive

Graph clustering, which is a type of hard clustering algorithm, can be further subdivided into two primary types as discussed by Aggarwal and Wang [2010]: node and graph clustering algorithms. While node clustering has been explained as algorithms to cluster nodes within a single graph, its counterpart focuses on algorithms to cluster nodes from several graphs.

2.4.1 Spectral Clustering Algorithms: NJW and Self-Tuning

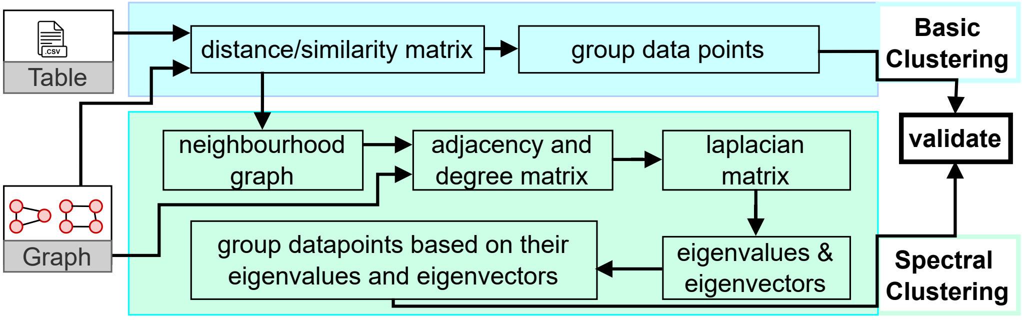


Table
Graph
distance/similarity matrix
group data points
Basic Clustering
validate
neighbourhood graph
adjacency and degree matrix
laplacian matrix
eigenvalues & eigenvectors
group datapoints based on their eigenvalues and eigenvectors
Spectral Clustering

Figure 2.8: Step-wise comparison of spectral against basic clustering algorithms.

Spectral clustering, which is a type of node clustering technique, differs from regular clustering through its use of the spectrum (eigenvalues) of a similarity matrix for partitioning data points (see figure 2.8). This is achieved by computing the Laplacian matrix from the adjacency and degree matrix. The generated Laplacian matrix is then used to generate eigenvalues and eigenvectors, of which k largest eigenvectors are selected to apply k-means clustering on, where k denotes the number of desired clusters or partitions. By choosing the k largest eigenvectors, we reduce the dimensionality of the data while retaining the most important information about the structure of the data. This makes it easier to cluster the data points based on their similarity in the low-dimensional space. In addition, choosing the k largest eigenvectors ensures that the resulting clusters have a meaningful structure that reflects the underlying geometry of the data. If we were to choose fewer eigenvectors, the clusters may not be well-defined, while choosing more eigenvectors would result in overfitting and reduced interpretability due to increased complexity.

Spectral clustering has gained the most popularity in terms of research and application when it comes to clustering nodes within a graph. It is often used when the data points cannot be easily separated by a linear boundary, and when there is a clear underlying geometric structure to the data. Regular clustering algorithms such as k-means, most often focus on minimizing the sum of squared distances between data points in the same cluster. However, this approach often fails to handle non-convex clusters which can be overcome through the use of spectral clustering. Several forms of it are available, with new subtypes being constantly developed to be applied from generic to specialized clustering applications. A comparison of spectral clustering against other popular clustering algorithms is shown in table 2.3.

Partitioning a graph can be achieved through two methods solving the following problems: the two-way ratio cut problem and the k-way ncut problem. The former focuses on solving the two-way ratio cut problem and dividing the graph into just two partitions. This is simple and fast, however, would not provide an optimal solution for complex graphs. The solution to the k-way ncut ratio problem solves this by partitioning the graph into k subsets, which is more computationally expensive.

Algorithm 2.4.3 NJW Algorithm [Ng et al., 2001]

```
Input: Graph G, its normalized Laplacian matrix $\mathrm{L_{sym}}$ and the number of desired clusters k.
Find k eigenvectors of the normalized Laplacian matrix L, arranging them in matrix U'.
Generate matrix U by normalising each row of U'.
Apply k-means algorithm on matrix U and find k partitions.
Assign nodes to clusters if their eigenvalue belongs to the partition.
Output: k partitions.
```

The first-ever use of eigenvectors for clustering was observed in the EIGI algorithm [Hagen and Kahng, 1992], where the authors explained the use of Fiedler eigenvectors to propose a solution to find the lower bound the two-way ratio cut problem [Nascimento and De Carvalho, 2011]. The Fiedler eigenvector is the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix of a graph. This resulted in an algorithm which could divide the data points into 2 partitions. The next major breakthrough in spectral clustering was proposed by Shi and Malik [2000], where the authors proposed a solution to the k-way ncut problem and this resulted in a spectral clustering algorithm which could provide k clusters instead of just two. Later Andrew Ng, Michael Jordan, and Yair Weiss proposed an improved solution (see algorithm 2.4.3) over Shi and Malik [Ng et al., 2001], where the algorithm normalized the graph Laplacian before generating eigenvectors. All of these algorithms use various functions to generate a similarity (affinity) score between pairs of data points. The NJW algorithm achieves this using the following equation:

$$A_{\mathrm{ij}} = exp(-||s_{\mathrm{i}} - s_{\mathrm{j}}||^2/2\sigma^2) \tag{2.3}$$

where, $\mathrm{A_{ij}}$ = affinity between points $\mathrm{s_i}$ and $\mathrm{s_j}$
and, $||\mathrm{s_i}\text{-}\mathrm{s_j}||$ = Euclidean distance between points $\mathrm{s_i}$ and $\mathrm{s_j}$

Equation 2.3 can be interpreted as the affinity of points $\mathrm{s_i}$ and $\mathrm{s_j}$, being calculated as the Euclidean distance between points $\mathrm{s_i}$ and $\mathrm{s_j}$, scaled down by a factor σ. However, this equation requires an optimal choice to be made for the values of σ to obtain good clustering results. Soon after, Zelnik-manor and Perona [2004] presented the term local scaling by introducing two scaling parameters in their equation and generate affinity between data points as shown in the following:

$$A_{\mathrm{ij}} = exp(-||s_{\mathrm{i}} - s_{\mathrm{j}}||^2/2\sigma_{\mathrm{i}}\sigma_{\mathrm{j}}) \tag{2.4}$$

The addition of another parameter seems to increase the complexity of the equation. However, local scaling provides a method to automatically detect an optimal value for σ_{i} by studying the local statistics of the neighbourhood of point $\mathrm{s_i}$ (or $\mathrm{s_i}$). In their research, they used the following equation to compute optimal σ_{i} (or σ_{i}):

$$\sigma_{\mathrm{i}} = ||s_{\mathrm{i}} - s_{\mathrm{K}}|| \tag{2.5}$$

where $\mathrm{s_K}$ = $\mathrm{K^{th}}$ neighbour point of $\mathrm{s_i}$. While observing the effects on synthetic and image data, Zelnik-manor and Perona [2004] used a value of K = 7.

Table 2.4: Comparison of spectral clustering algorithms [Nascimento and De Carvalho, 2011]

	Complexity	Laplacian	Application
EIGI	O(n^2)	unnormalized	- load balancing
MELO	O(n^2d)		- general clustering
Anchor	O(nmd)	non-Laplacian	- text and document categorization
Shi and Malik	O(nmd)	normalized	- image segmentation
Ng-Jordan-Weiss	O(nmd)		- dimension reduction
Self-tuning	O(nmd)		- general clustering

It is evident from table 2.3 that spectral clustering is computationally expensive, similar to hierarchical clustering. However, spectral clustering is highly effective when handling complex metaproteomic data which most often contain a high number of columns (metaproteins). In table 2.4, I have compared them against a few other spectral clustering algorithms before discussing them in detail.

Although, during the conception of early algorithms like EIGI and MELO, spectral clustering was mostly used for applications such as electrical load balancing and general-purpose clustering. However, with the emergence of algorithms by Shi and Malik [2000], Ng et al. [2001] and Zelnik-manor and Perona [2004], spectral clustering became widely popular for image segmentation as well as a dimension reduction technique, as shown in table 2.4.

2.4.2 Clustering Evaluation: Internal and External Indices

Clustering evaluation metrics are used to measure the quality of clusters and the effectiveness of clustering algorithms. These metrics can be broadly classified into three categories: internal, external and relative indices. Internal and external indices can be regarded as statistical measures whereas relative indices are non-statistical. Several internal, external and relative indices for clustering have been compiled and compared in table 2.5.

Internal indices are used to evaluate the clustering results based on the characteristics of the data itself e.g., , the similarity between data points. They don't require any external information, such as the true cluster labels, to evaluate the clustering performance. Some commonly used internal indices are:

- **Silhouette score** measures how similar are points in the same cluster and how different they are in neighbouring clusters Günter and Bunke [2003].

  $$Silhouette_{\mathrm{i}} = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{2.6}$$

  where a(i) = average intra-cluster distance between i and every other point in the same cluster as i (separation),
  and b(i) = the minimum average distance between i and every other point in a different cluster than i (cohesion).

- **Davies-Bouldin index** measures the average similarity between each cluster and its most similar cluster, relative to the average dissimilarity between each cluster and its most dissimilar cluster Davies and Bouldin [1979].

$$DB = \frac{1}{k}\sum_{i=1}^{k} max_{j\neq i}(D_{i,j}) \tag{2.7}$$

$$D_{i,j} = \frac{(\bar{d_i} + \bar{d_j})}{d_{i,j}} \tag{2.8}$$

  where, DB = Davies-Bouldin index, $\overline{d_{\mathrm{i}}}$ and $\overline{d_{\mathrm{j}}}$ are the average distance between each point in the i^{th}, and the j^{th} cluster respectively and $\mathrm{d}_{\mathrm{i,j}}$ is the Euclidean distance between the centroids of the i^{th} and the j^{th} cluster.

External indices require external information e.g., true cluster labels. Such metrics compare the clustering results with the ground truth labels and are often used for validation and comparison of various clustering algorithms e.g., :

- **Adjusted rand index** measures the similarity between the true labels and the predicted labels. It is a weighted form of the rand index/statistics Halkidi and Vazirgiannis [2001].

$$ARI = \frac{RI - ExpectedRI}{max(RI) - ExpectedRI} \tag{2.9}$$

  where ARI = Adjusted Rand Index, and RI = Rand Index

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.10}$$

  where TP = True Positive (same class and same cluster), FN = True Negative (same class and different clusters), FP = False positive (different class and same cluster), and TN = False Negative (different class and different clusters)

- **Fowlkes-Mallows index** measures the geometric mean of the precision and recall 2.5.2 of the predicted labels relative to the true label Halkidi and Vazirgiannis [2001].

$$FMI = \frac{TP}{\sqrt{(TP + FP) * (TP + FN)}} \tag{2.11}$$

  where, FMI = Fowlkes-Mallows index

Relative indices are used mostly in the context of fuzzy clustering e.g., Figure of Merit (FOM) stability and are out of scope for spectral clustering evaluation. A list of possible clustering evaluation indices is categorized in table 2.5 with their types and clustering types they are used for. In the following table 2.5, I have compiled and compared several clustering evaluation metrics.

Table 2.5: Clustering validation indices [Gan et al., 2007] [Brun et al., 2007].

		Clustering validation indices						
Statistical	**External indices**	Dunn's indices [Dunn, 1973]	Rand statistic [Halkidi et al., 2001]	Jaccard coefficient [Halkidi et al., 2001]	Folkes and Mallow index [Halkidi et al., 2001]	Hubert's T statistic [Theodoridis and Koutroumbas, 1999]	Normalized T statistic [Halkidi et al., 2002]	**Hierarchical and Hard (Partitional) clustering index**
	Internal indices		Cophenetic correlation coefficient [Farris, 1969]					
		Root- mean-square standard deviation [Sharma, 1996]	Davis-Bouldin index [Davies and Bouldin, 1979]	SD index [Halkidi et al., 2000]	S_Dbw index [Halkidi and Vazirgiannis, 2001]	Silhouette index [Günter and Bunke, 2003]		
			Root Squared index [Sharma, 1996]	Calinski-Harabasz index [Calinski and Harabasz, 1974]	Semi. partial R-squared [Sharma, 1996]	Average of compactness [Zaït and Messatfa, 1997]	Distance between partitions [De Mántaras, 1991]	
Non-statistical	**Relative indices**	Figure of merit (FOM) [Yeung et al., 2001]			Stability [Lange et al., 2002]			
		Partition coefficient index [Bezdek, 1987]	Partition entropy index [Bezdek, 1973]	Fukuyama-Sugeno index [Fukuyama and Sugeno, 1989]	based on fuzzy similarity [Jihong and Xuan, 2000]	Fuzzy validity criterion [Xie and Beni, 1991]	Partition separation index [Yang and Wu, 2001]	**Fuzzy clustering index**

2.5 Supervised Learning: Classification

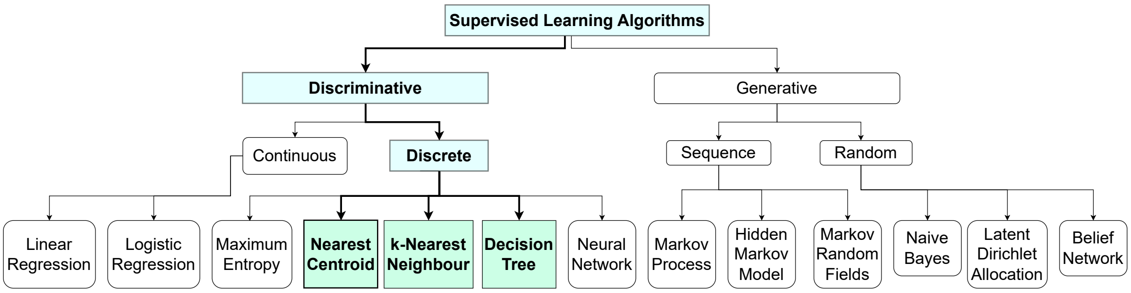


Figure 2.9: Taxonomy of classification algorithms [Nicolas, 2015].

Machine learning models for categorization tasks typically fall into two broad categories: discriminative and generative classifiers (see figure 2.9). They use different approaches to the challenge of learning to classify incoming data. A discriminative classifier immediately learns where one class ends and another begins without directly modelling the underlying probability distributions of the input data. A generative classifier simulates the combined probability distribution of the input characteristics and the output variable. It gains the ability to affect the prior probability for each class as well as the probability distribution of the input characteristics for each class. The conditional probability of the output variable given the input features can then be calculated using Bayes' rule.

Generative supervised learning algorithms can be further categorized into sequence generative models and random generative models, based on how they generate new examples. Sequence generative models are used for tasks that involve generating sequences of output labels, such as speech recognition, natural language processing, and handwriting recognition. Examples include Hidden Markov Models (HMM), Recurrent Neural Networks (RNN), and Conditional Random Fields (CRF). Random generative models are used for tasks that involve generating new examples of input features and output labels that are similar to those in the training data, but not necessarily in a specific order or sequence. Examples include Gaussian Mixture Models (GMM), Naive Bayes, and Generative Adversarial Networks (GAN).

Discriminative supervised learning algorithms can be further categorized into continuous and discrete algorithms, based on the nature of the output labels. Continuous discriminative algorithms are used for tasks that involve predicting continuous output values. Examples include Linear Regression, Logistic Regression etc. Discrete discriminative algorithms are used for tasks that involve predicting discrete output labels, such as classification problems. Examples include Naive Bayes, k-Nearest Neighbors (k-NN), Decision Trees, Random Forests, Gradient Boosted Trees, and Neural Networks.

2.5.1 Classification Algorithms: Nearest Centroid, k-Nearest Neighbour and Decision Tree

In table 2.6, I have compared the three discrete discriminative classification algorithms used in this research: nearest centroid, k-nearest neighbour (k-nn) and decision tree algorithms, for further discussion.

Table 2.6: Comparison of classifying algorithms [Alpaydin, 2020].

	Type	**Effective on**	**Complexity**	**Noise Handling**
Nearest Centroid	Parametric	small data	O(nd) [Levner, 2005]	inferior
k-NN	Non-parametric	small data	O(nmd) [Kuang and Zhao, 2009]	inferior
Decision Tree	Non-parametric	large data	O(nd*logn) [Kuang and Zhao, 2009]	superior

A quite underrated and overlooked classifier is the nearest centroid algorithm (algorithm 2.5.1) which classifies new data points based on their similarity to the centroid of the same class in the training data. A centroid is the most average representative for a given set of points or coordinates. In terms of data, the centroid is calculated by summing all values for each individual feature and dividing by the number of data points. In order to determine the centroid of every class, the algorithm averages the feature values of all training examples in the class. A new point is then classified by finding the closest centroid and assigning the corresponding class label.

Algorithm 2.5.1 Nearest Centroid Algorithm [Forgy, 1965]

```
1: Input: Dataset containing training samples as rows and features as columns, and
   labels for each sample.
2: Training:
3: Calculate the centroid for each class as the mean vector of all training samples
   for any given class.
4: Testing:
5: For each new sample, calculate the distance to all centroids.
6: Assign class label of the nearest centroid.
7: Repeat steps 5 and 6 for all samples in the test dataset.
8: Output: Predicted labels for test data.
```

The nearest centroid algorithm is fast and easy to implement however does not perform well on high-dimensional data. Similar to k-means, the algorithm divides up the training samples into equal-sized partitions which also makes it a bad choice for data representing a non-equal-shaped distribution of class labels. However, when it comes to the eigenvectors of any given data, logically nearest centroid classifier should be efficient enough to partition the low-dimensional representation of the original data. Being based on the concept of centroid, the nearest centroid classifier assumes that training and testing data is normally distributed, which makes it a parametric classifier.

k-NN (algorithm 2.5.2) has a similar approach to the nearest centroid and classifies new data points based on the class labels of their nearest neighbour in the training data. The value of k is a hyperparameter that is required to be specified before training the algorithms. In order to categorize a new data point, the distance between the new data point and all the other data points in the training set must first be calculated. Then, k nearest neighbours must be chosen, and the label for the class with the highest number of nearest neighbours is then assigned.

Algorithm 2.5.2 k-Nearest Neighbour Algorithm [Cover and Hart, 1967]

```
1: Input: Dataset containing training samples as rows and features as columns,
   labels for each sample and desired number of closest neighbours k.
2: Testing:
3: For each new sample, calculate the distance to all training samples.
4: Select k samples with the smallest distances to the test sample.
5: Assign label to test sample with the most frequent label among k selected samples.
6: Repeat steps 3, 4 and 5 for all samples in the test dataset.
7: Output: Predicted labels for test data.
```

k-NN is a lazy learning algorithm, meaning that it does not have a traditional training phase like other classifiers. Instead, the training samples are simply stored and then labels are assigned during the testing phase of the algorithm. Additionally, k-NN is non-parametric, which means it does not assume any underlying probability distribution of the data, making it adaptable to different types of data.

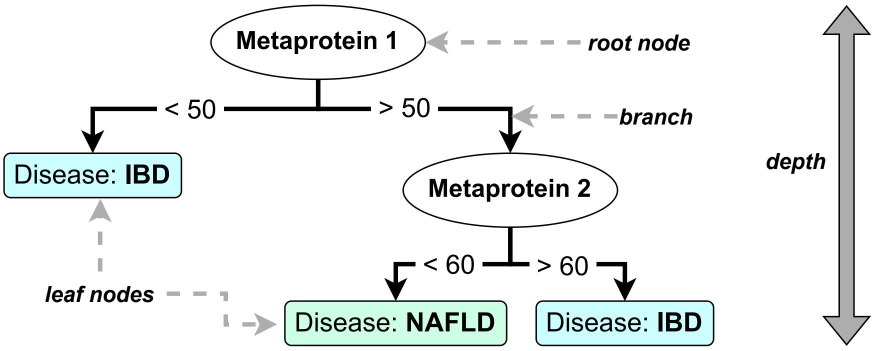


Figure 2.10: Components of a decision tree. The depth is 3 in this case.

The decision tree on the other hand builds a tree-like model of decisions and their possible consequences. The internal nodes of the tree correspond to decisions based on input features and the leaf nodes represent the class labels. When a new data point is to be classified, the algorithm traverses the tree from the root node to the leaf node based on its feature values.

k-NN and decision tree algorithms are non-parametric, meaning that they do not make any assumption about the underlying distribution of the data. Contrarily, the nearest centroid is parametric as it assumes data is normally distributed with class-specific means. Another important consideration is the size of the dataset each algorithm is effective on. k-NN and the nearest centroid is best suited for small datasets and the former is faster in terms of speed of execution. A decision tree is more effective on medium to large datasets. Furthermore, each algorithm handles noise differently. Nearest centroid and k-nn are sensitive to noise in the data while the decision tree can handle outliers effectively because the splitting criterion is not based on specific values of individual data points. In figure 2.11, I have provided a

Algorithm 2.5.3 Decision Tree Algorithm [Li et al., 1984] [Quinlan, 1986]

```
1: Input: Dataset containing training samples as rows and features as columns, and labels for each sample.
2: Training:
3: Select the most informative feature or attribute for splitting the root node, using a criterion such as information gain or the Gini index.
4: Partition training samples into subsets based on the selected feature, with each subset corresponding to a child node of the root.
5: Repeat steps 3 and 4 for each child node till a stopping criterion is met or until all samples in any leaf node belong to the same class.
6: Assign a class label to each leaf node based on the majority class of all the samples in the child node.
7: Testing:
8: For each new sample, compare the value of the feature at each decision node and follow the branch till a child node is reached.
9: Assign class label associated with leaf node reached.
10: Repeat steps 8 and 9 for all samples in the test dataset.
11: Output: Predicted labels for test data.
```

visualisation of how the decision boundaries differ across the three algorithms when trained on the iris dataset in figure 2.11 [2].

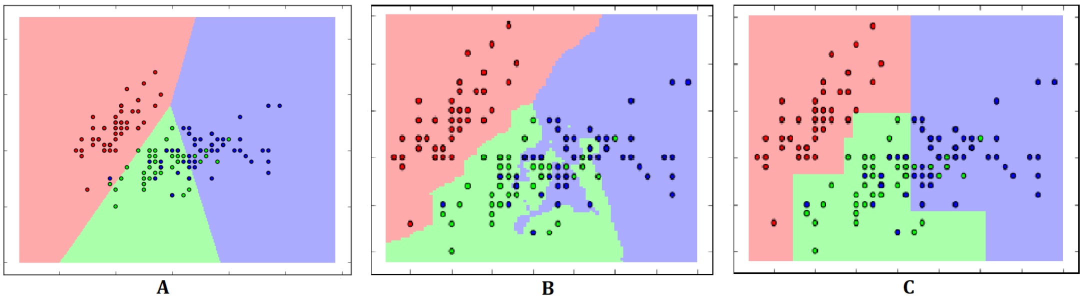


Figure 2.11: Decision boundaries after training on the iris dataset for A)Nearest Centroid B) k-Nearest Neighbour C) Decision Tree (Borrowed figures)

The decision boundaries explicitly reflect one of the major differences between the three classifiers. The nearest centroid classifier has straight linear decision boundaries making it ideal for very simple datasets with labels separated at equal distances. However, the k-nn has an adaptive non-linear decision boundary making it suitable for more complex data types. On the other hand, the decision boundary of a decision tree is a set of linear or axis-parallel decision rules that divide the feature space into rectangular regions. The Decision Tree decision boundary is typically piece-wise constant and can only take the form of rectangles, parallelograms, or hyperplanes. A comparison between these types of classifiers implemented on metaproteomic

[2] **Sources:** http://stephanie-w.github.io/brainscribble/classification-algorithms-on-iris-dataset.html
https://scikit-learn.org/0.17/auto_examples/neighbors/plot_nearest_centroid.html

abundance would help us realize the optimal type of decision boundaries suitable when dealing with such data against the eigenvectors of the abundances.

2.5.2 Classification Evaluation

After training a machine learning model on a classification task, evaluation metrics assess the performance of the model. Therefore, unseen data (test data) are passed to a machine learning model and the predicted labels are compared to the true labels. The comparison of predicted labels against true labels leads to four different scenarios that can be represented in a confusion matrix as shown in table 2.7.

Table 2.7: Confusion matrix for comparing predicted and actual labels.

		Actual	
		True	**False**
Predicted	**True**	True Positive	False Positive
	False	False Negative	True Negative

Most classifier evaluation metrics could be derived from the corresponding confusion matrix in order to evaluate and improve the performance of the model. Some popular classifier evaluation metrics are discussed below [Chicco and Jurman, 2020]:

- **Accuracy** measures the proportion of correct predictions made by the model and is one of the most widely used metrics for the evaluation of predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.12}$$

  where TP = True Positive, TN = True Negative, FP = False positive, and FN = False Negative.

- **Precision (Positive Predictive Value)** = TP/(TP+FP), measures the proportion of true positive predictions out of all the positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{2.13}$$

- **Recall (True Positive Rate)** measures the proportion of true negative predictions out of all the actual positive instances.

$$Recall = \frac{TP}{TP + FN} \tag{2.14}$$

- **Matthews Correlation Coefficient (MCC)** takes into account all four values in the confusion matrix and is especially useful when there is a class imbalance in the data.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \tag{2.15}$$

  where TP = True Positive, TN = True Negative, FP = False positive, and FN = False Negative.

3. Related Work

In this section, I discuss the related work regarding clustering applications in multi-omics data, especially in metaproteomics. I have used the datasets from two of these researches for implementing and evaluating spectral clustering and classification. In the first section, I have discussed related work regarding the clustering of omics data. And later, in the second section, I have mentioned related work in the context of implementing spectral clustering on omics data.

Table 3.1: List of related work discussed in this section.

	Description	Type of Data	Domain
Metaproteomics of gut microbiome of patients with IBD [Lehmann et al., 2019]	Hierarchical clustering on metaproteomic abundances.	Metaproteomic abundance	metaproteomics
SpectralTAD [Cresswell et al., 2020]	Clustering framework that uses gaps between consecutive eigen vectors.	Hi-C data of genomes	genomics
Spectrum [John et al., 2020]	Spectral clustering method for complex omic data	Single- and multi-omic data	omic
Metaproteomics of gut microbiome of patients with NAFLD [Sydor et al., 2022]	Hierarchical clustering and logistic regression on metaproteomic abundances.	Metaproteomic abundance	metaproteomics

There have been several applications of spectral clustering in various domains of omics data analysis such as genomics and metaproteomics. In table 3.1, I have compiled a list of literature, that has either used spectral clustering as part of the data processing pipeline or modified spectral clustering algorithm to make it a better fit for their purpose.

3.1 Clustering of Omics Data

Research on the clustering of multi-omics data has increased significantly in recent years. Organizing samples or features that display comparable trends across various omics data types is the aim of clustering in this context. This can aid in the identification of biologically pertinent groups that might be linked to particular phenotypes or disease conditions. Researchers can find biomarkers that are differentially expressed between groups by grouping samples based on their omics profiles. These biomarkers can then be used to create diagnostic or prognostic tests or to find possible therapeutic targets.

3.1.1 Metaproteomics of the gut microbiome of patients with IBD

In one such study by Lehmann et al. [2019], the gut microbiomes of patients with Crohn's disease (CD) and ulcerative colitis (UC), two inflammatory bowel diseases (IBD) that impact the gastrointestinal tract, were compared. In the research, gut microbiomes from 17 healthy controls, 11 CD patients, 14 UC patients, 13 Irritable Bowel Syndrome (IBS) patients, 8 Colon Adenoma (CA) patients, and 8 Gastric Carcinoma (GCA) patients were investigated. The samples' proteins were identified and measured using non-invasive LC-MS/MS, and variations in protein expression between the three groups were found by analyzing the data. After protein identification, hierarchical clustering was applied to the abundance values to group similar patients and identify disease-specific metaprotein patterns.

As a result, healthy subjects were distinguished from patients with CD and UC as well as from patients with GCA using cluster analysis and non-parametric test (analysis of similarities). Furthermore, the results showed that the protein expression profiles of patients with Crohn's disease were considerably different. In particular, they found that individuals with Crohn's disease had a higher abundance of proteins related to oxidative stress, immune response, and inflammation. In contrast to both Crohn's disease patients and healthy controls, they also discovered that ulcerative colitis patients had greater levels of proteins involved in energy metabolism and nutrient transport.

These findings suggest that metaproteomics can be a valuable tool for understanding the complex interactions between the gut microbiome and the host in inflammatory bowel diseases. By identifying the specific proteins that are dysregulated in these diseases, researchers can develop targeted therapies These findings suggest that metaproteomics can be a valuable tool for understanding the complex interactions between the gut microbiome and the host in inflammatory bowel diseases. By identifying the specific proteins that are dysregulated in these diseases, researchers can develop targeted therapies to treat oxidative stress and inflammation, as possible examples.

3.1.2 Metaproteomic of the gut microbiome of patients with NAFLD

In another similar work on the human gut microbiome by Sydor et al. [2022], the gut microbiomes of patients with non-alcoholic steatohepatitis (NASH) and hepatocellular carcinoma (HCC) to identify potential biomarkers for NASH patients with and without HCC. NASH and HCC are two non-alcoholic fatty liver diseases (NAFLD) that impact the liver. The research, which aimed to identify diagnostic biomarkers, included 19 healthy controls, 32 NASH patients, and 29 HCC patients. Hierarchical clustering with Canberra distance was applied to metaproteomic abundance values to group similar patients. Additional analysis was performed to identify differences in the protein profiles between the two groups.

The researchers identified 155 differentially abundant proteins between NASH patients with and without HCC. These proteins were engaged in a number of biological processes, such as lipid metabolism, immune response, and inflammation. In earlier research, several of these proteins were also linked to HCC, highlighting their potential as biomarkers for early detection. Sydor et al. [Sydor et al., 2022] also suggest a possible function of the gut microbiome in the development of HCC in NASH patients due to an increased abundance of bacterial species in NASH patients with HCC. However, the study could not identify any single bio-marker to separate NASH and HCC. Nevertheless, the distinction between controls, NASH, and HCC could be made with an accuracy of 86% using machine learning-based classification methods.

Overall, similar to the work described in the previous section, the research suggested that metaproteomic analysis can enable early detection of NASH.

3.2 Spectral Clustering on Omics Data

The large dimensionality of multi-omics data presents itself as a difficulty in the analysis of multi-omics data. Spectral clustering can overcome this problem by reducing the dimensionality of multi-omics data while preserving the most important information/features. Additionally, biomarkers linked to particular patient clusters can be found using spectral clustering. These biomarkers can shed light on the variations in biological processes between analyzed groups.

3.2.1 SpectralTAD

Chromatin conformation capture techniques [3] like Hi-C can identify discrete, self-interacting genomic regions known as topologically associated domains (TADs). SpectralTAD is a spectral clustering framework that utilizes the gap between consecutive eigenvectors for boundary identification of topologically associated domains (TAD). By dividing genomic regions into functionally separate compartments and by modifying the interactions between regulatory elements and their target genes, TADs are thought to be essential for controlling gene expression.

[3] Chromatin conformation capture (3C) techniques are a family of molecular biology methods that are used to study the three-dimensional organization of the genome inside the cell nucleus.https://epigenie.com/epigenetics-research-methods-and-technology/chromatin-analysis/chromatin-conformation-analysis-3c-techniques/

In this research by Cresswell et al. [2020], a package called SpectralTAD was created within the programming framework R, for Hi-C data. It can identify hierarchical, biologically relevant TADs and has automatic parameter selection. In both simulated and real-world situations, SpectralTAD outperforms four cutting-edge TAD analysers. The research also showed that TAD boundaries, shared between various layers of the TAD hierarchy, were more enriched in traditional boundary marks and more conserved between different cell types and tissues. TADs that cannot be divided into sub-TADs, however, exhibited less enrichment and conservation at their boundaries, indicating a more dynamic function in genome regulation.

3.2.2 Spectrum

In this research by John et al. [2020], a new algorithm called ”Spectrum” is proposed for clustering single and multi-omic data. Spectrum uses a self-tuning density-aware kernel which is a type of kernel function that takes into account the density distribution of data points when calculating similarity. In addition to noise reduction and revealing the underlying structure of input data, Spectrum also consists of a new method to find the optimal number of clusters k, by analysing the distribution of eigenvectors. Spectrum has provided competitive results when tested on seven different single and multi-omic datasets, outperforming several existing algorithms e.g., both in terms of accuracy and speed.

4. Methodology

In this chapter, I will portray and discuss the data and methods used in this research to compare clustering and classification tasks, executed on three datasets. The first section describes each dataset used in detail. In the second section, I discuss the data transformations used in the experiments. In the thrid and fourth sections, in the last two sections, I have outlined the structure of the two experiments in detail. And finally, in the last section, I have outlined the hardware and software details of the experimental setup.

In figure 4.1, a brief overview of the thesis methodology has been outlined. It can be segmented into four main parts:

- **Data selection and pre-processing.**
- **Data transformation.**
- **Experiment 1:** Comparison of clustering between agglomerative, different types of spectral clustering: using k-nearest neighbour and complete graph (Ng-Jordan-Weiss and self-tuning algorithm), for different datasets and for different transformations.
- **Experiment 2:** Comparison of classification between nearest centroid, k-nn and decision tree classifiers, for different datasets and different transformations.

4.1 Data Selection and Pre-processing

In table 4.1, I have compared the used datasets and it is evident that the most significant changes are observed in the number of metaproteins observed for each dataset. Dataset 1 and dataset 2 have been derived from original research whereas dataset 3 is a combination of both. All the data used in this research were metaproteomic abundance data from the human gut of diseased and control patients. In the following, for each dataset, I have discussed the source research papers and the pre-processing steps that I have applied for my experiments.

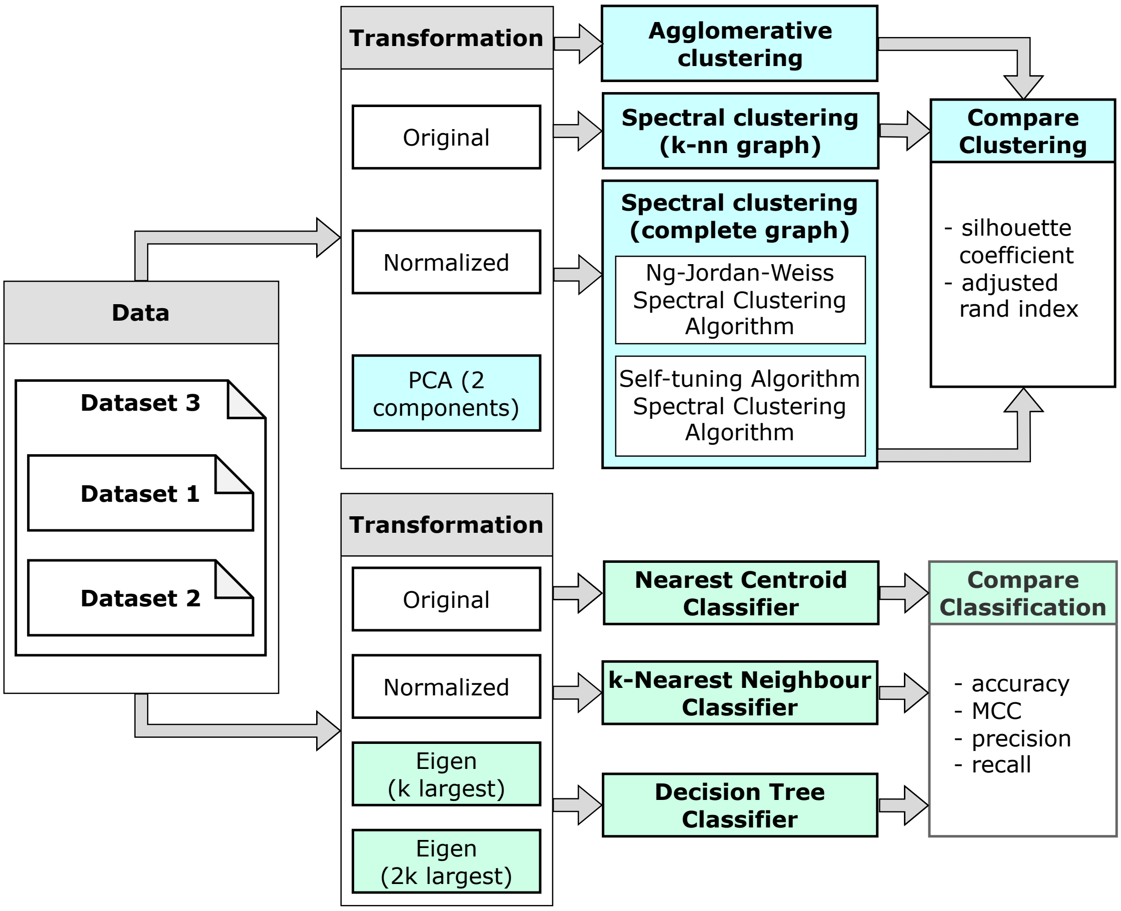


Figure 4.1: Methodology for this research. Arrows represent the data flow. Blue boxes denote events used for the clustering tasks and green boxes represent events used for classification tasks only. MCC = Matthew's Correlation Coefficient.

Table 4.1: Comparison of datasets used in this research. IBS = Inflammatory Bowel Diseases, NAFLD = Non-Alcoholic Fatty Liver Diseases.

	Data type	Source research goal	No. of metaproteins		No. of patients	No. of labels
			actual	selected		
Dataset 1 [**Lehmann et al., 2019**]	gut metaproteomic abundance	discover biomarkers of IBS	2969	1752	76	7
Dataset 2 [**Sydor et al., 2022**]	gut metaproteomic abundance	discover biomarkers of NAFLD	42574	10	80	3
Dataset 3 (Datasets 1 + 2)	gut metaproteomic abundance	not applicable	170	170	156	9

4.1.1 Dataset 1: Metaproteomics of faecal samples of patients with Inflammatory Bowel Diseases.

This dataset was retrieved from a study regarding inflammatory bowel diseases (IBD) where mass spectrometry (LC-MS/MS) was used to observe disease-specific microbial and human proteins in faecal samples taken from patients. Using a metaproteomic approach, this research examined the effect of the gut microbiome on IBS diseases, especially Crohn's Disease (CD) and Ulcerative Colitis (UC) and identified several disease-specific marker proteins. Their work [Lehmann et al., 2019] suggested that faecal metaproteomics is a helpful non-invasive tool for a more accurate and straightforward diagnosis of both diseases, CD and UC.

Even though the focal diseases were CD and UC, in their originally observed data, there are seven class labels: control, CD, IBS, Ulcerative colitis in remission (UCr), active Ulcerative Colitis (UCa), Gastric Carcinoma (GCA) and Colon Adenoma (CA) patients. A high number of observed metaproteins, in this case, is already a large contributor to increasing computational complexity and time. As a result. for this research, only human and microbial metaproteins were chosen for analyses, reducing the number of columns from 2,969 to 1,752. This step also removed metaprotein columns that had very low average abundance for each patient e.g. columns with mostly zero values. As a result, in my thesis, 76 patients were observed in this dataset for 1,752 metaproteomic abundances from samples consisting of 7 distinct class labels.

Lehmann et al. [2019] performed hierarchical clustering on dataset 1, providing a benchmark comparison only for the clustering results. In this thesis, I have performed spectral clustering on this dataset to analyze which clustering methods perform better. Furthermore, I have performed classification to compare the ability of eigen (k and 2k), normalization and no transformation, to predict the disease of patients from the given metaproteomic abundances for each patient.

4.1.2 Dataset 2: Discovering Biomarkers for Non-Alcoholic Steatohepatitis Patients using Fecal Metaproteomics

This dataset was retrieved from a study regarding non-alcoholic fatty liver diseases (NAFLD), where mass spectrometry (LC-MS/MS) was used to find diagnostic biomarkers from metaprotein abundances observed in faecal samples, taken from patients suffering from non-alcoholic steatohepatitis (NASH) or hepatocellular carcinoma (HCC). To find diagnostic indicators, Sydor et al. [2022] examined the faecal metaproteome of 19 healthy controls, 32 NASH patients, and 29 HCC patients. They observed that NASH and HCC caused changes in the gut microbiome's composition, resulting in a rise in the inflammation of the gut. Their work also showed that although a single biomarker was unable to differentiate between NASH and HCC, however, machine learning-based classification algorithm (5-fold cross-validation of Linear Discriminant Analysis, for 10,000 iterations) was able to do so with an accuracy of 86% using several biomarkers, thus proving that faecal metaproteomics offers early identification of NASH and HCC.

The original data of this research had an unusually high number of 42,574 metaproteins that were observed. Furthermore, the data contained many metaproteins whose mean abundance for all the patients was mostly observed as 0 or significantly close to 0. As a result, to reduce computational expenses, only 10 metaproteins, which were observed as potential biomarkers in the original research, were chosen for this thesis. This made dataset 2 sufficiently less complex and less computationally expensive to analyse and observe. As a result, in my thesis, 80 patients were analysed in this dataset for 10 metaprotein abundances from samples consisting of 3 distinct class labels: control, NASH and HCC.

Sydor et al. [2022] has used dataset 2 for both clustering and classification tasks in the experiments of this research. In the original research, it has been used for both clustering and classification tasks as well as providing benchmark comparisons for both experimental results.

4.1.3 Dataset 3: Combining Dataset 1 and Dataset 2

Dataset 3 is a combination of datasets 1 and 2 which was created to investigate the effect of merging patients of several diseases with respect to the common metaproteins, found in both sets of samples. Several metaproteins can be measured in all patients because all humans share a set of metaproteins that are responsible for similar biological processes. However, there can still appear large differences in the metaproteome profile of different patients due to different living and health conditions. One major issue when dealing with metaproteomic data arises from the presence of several thousand metaproteins for a handful of patients. Both datasets 1 and 2 contain less than a hundred patients each, with several thousand observed metaproteins for each of them. This results in a high number of features for clustering and classification algorithms when it comes to grouping and predicting labels of patients, respectively.

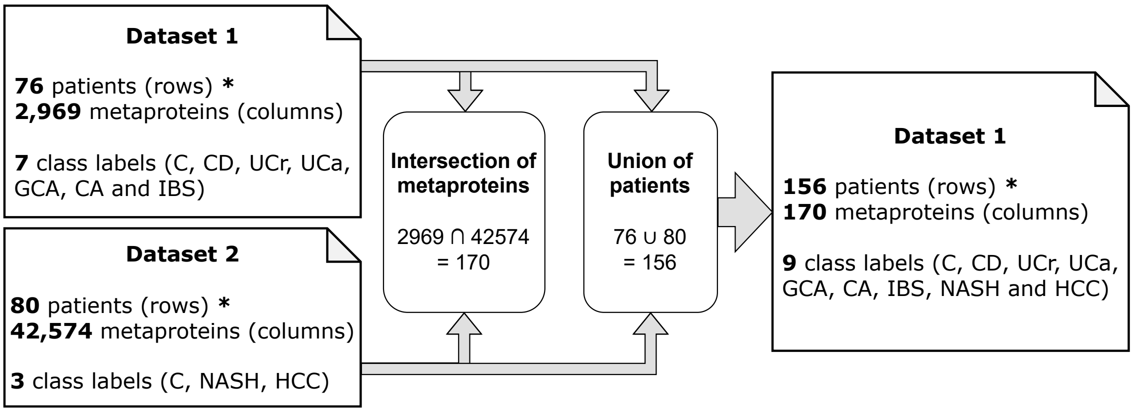

Dataset 1
76 patients (rows) *
2,969 metaproteins (columns)
7 class labels (C, CD, UCr, UCa, GCA, CA and IBS)
Dataset 2
80 patients (rows) *
42,574 metaproteins (columns)
3 class labels (C, NASH, HCC)
Intersection of metaproteins
2969 ∩ 42574 = 170
Union of patients
76 ∪ 80 = 156
Dataset 1
156 patients (rows) *
170 metaproteins (columns)
9 class labels (C, CD, UCr, UCa, GCA, CA, IBS, NASH and HCC)

Figure 4.2: Flowchart to elaborate the formation of dataset 3. Arrows represent the data flow.

The following steps were taken to merge datasets 1 and 2 and create dataset 3 as depicted in figure 4.2:

1. **Intersection of columns (Finding common metaproteins)**: Of the 1,752 metaproteins found in dataset 1 and 12,649 in dataset 2, I found 170 metaproteins that were common for all patients in both datasets 1 and 2.

2. **Union of rows (Adding both sets of patients)**: 76 patients in dataset 1 and 80 patients in dataset 2 were combined to create 156 patients in dataset 3, for which the 170 common metaproteins were observed.

In addition to increasing the number of rows and contrary to decreasing the number of columns in dataset 3, the number of class labels increased to 9. This is due to the fact that of the 7 class labels in dataset 1 and 3 class labels in dataset 2, there exists a common label between the two datasets: control patients. As a result, 156 patients were analysed in dataset 3 with 170 metaproteins and 9 distinct class labels: control, CD, IBS, UCr, UCa, GCA, CA, NASH and HCC.

4.2 Data Transformation: Normalization, Eigendecomposition and PCA

As explained in chapter 2, two transformation techniques were applied for the clustering tasks:- normalization and principal component analysis (PCA) and two for the classification tasks:- normalization and eigendecomposition (k and 2k). Original data (no transformation) was kept as a control to compare with. After experiment 1 was performed, PCA seemed like a redundant step, as PCA has eigendecomposition within itself. As a result, for classification tasks, PCA was not used, since two eigendecompositions may possibly oversimplify important relationships between features.

Additionally, in the second experiment, I checked the efficiency of the eigendecomposition, used in the NJW and self-tuning spectral clustering algorithms, for classification tasks. To investigate this elaborately, I used k and 2k largest eigendecomposition features for each dataset, where k represents the number of class labels in the dataset. As a result, for dataset 1, the largest 3 and 6 eigendecomposition features were used for this transformation. For dataset 2, the largest 7 and 14 eigendecomposition features were used. And for dataset 3, the largest 9 and 18 eigendecomposition features were used.

4.3 Experiment 1: Comparison of Clustering

Algorithm 4.3.1 intends to illustrate experiment 1, the goal of which was to compare the quality of clustering, produced by the implemented algorithms. Having separate lists for the 3 datasets, 3 data transformations (including no transformation) and 4 algorithms, I generated one set of clustering results for each combination and evaluated the results. As a result, 36 clustering results were generated, and for each 2 evaluation metrics were measured: silhouette coefficient and adjusted rand index.

In this experiment, 3 clustering algorithms were compared: agglomerative and 3 types of spectral clustering. For agglomerative (hierarchical) clustering, the python package used is **sklearn.cluster.AgglomerativeClustering()**. The first type SC(package) uses the python package **sklearn.cluster.SpectralClustering()**, which by default creates a k-nn graph before performing eigendecomposition and k-means. The second and third types of spectral clustering were generated using eigendecompositions of the

Algorithm 4.3.1 Experiment 1

```
datasets = [dataset 1, dataset 2, dataset 3]
transformations = [original, normalized, PCA (2 components)]
algorithms = [agglomerative, spectral (k-nn), spectral (NJW), spectral (self-tuning)]
for $data \in datasets$ do:
    for $transform \in transformations$ do:
        for $algorithm \in algorithms$ do:
            Perform clustering.
            Measure silhouette coefficient.
            Measure adjusted rand index.
        end for
    end for
end for
```

Ng-Jordan-Weiss algorithm and self-tuning algorithms respectively, which derive the eigenvectors and eigenvalues from a fully connected graph. However, the NJW algorithm requires a parameter input, σ, from the user to calculate edge weights to represent the similarity between data points. This is solved by the self-tuning spectral clustering algorithm which automatically derives an optimal value for the parameter through local scaling. Both the NJW and self-tuning eigendecompositions were then clustered using the k-means algorithm. The python codes for NJW and self-tuning algorithms were derived from a blog by Sun [2020] where elaborate illustrations of the algorithms were provided with examples.

For this research, I have used the silhouette index/score/coefficient as the internal indices and the adjusted rand index as the external indices to evaluate spectral clustering algorithms and their resulting partitions. The silhouette index helped to comparatively evaluate spectral clustering algorithms based on the cluster separation when represented in the eigenspace. On the other hand, the adjusted rand index was helpful in realizing the efficiency of the eigenspace representation in capturing the underlying structure of the data with respect to the class labels. This was important to understand in order to apply the eigendecomposition as a pre-processing for predictive analytics on metaproteomic abundance data. Overall, experiment 1 has helped to compare clustering performance for the used transformation techniques, between hierarchical and spectral clustering, as well as between using a k-nn graph and a fully connected graph. The results would help identify optimal pathways while applying spectral clustering to group patients from metaproteomic abundances.

4.4 Experiment 2: Comparison of Classification

Experiment 2 is divided into two parts, the larger part is described in algorithm 4.4.1, where all three datasets are used. In the smaller subset of experiment 2 (algorithm 4.4.2), 5-fold cross-validation was applied to dataset 2, as was performed in the original research, to compare with the benchmark values.

Algorithm 4.4.1 portrays experiment 2a, the goal of which was to compare the quality of classification, produced by the algorithms - nearest centroid classifier, k-nn

Algorithm 4.4.1 Experiment 2a

```
datasets = [dataset 1, dataset 2, dataset 3]
transformations = [original, normalized, eigen (k), eigen (2k)]
algorithms = [nearest centroid, k-nn, decision tree]
for $data \in datasets$ do:
    for $transform \in transformations$ do:
        for $algorithm \in algorithms$ do:
            Perform classification.
            Measure accuracy for all labels.
            Measure Matthews Correlation Coefficient for all labels.
            Measure precision for label control.
            Measre recall for label control.
        end for
    end for
end for
```

Algorithm 4.4.2 Experiment 2b (Benchmark comparison for Dataset 2.)

```
datasets = [dataset 2]
transformations = [original, normalized, eigen (k), eigen (2k)]
algorithms = [nearest centroid, k-nn, decision tree]
for $data \in datasets$ do:
    for $transform \in transformations$ do:
        for $algorithm \in algorithms$ do:
            Perform 5-fold cross-validation for 10,000 iterations.
            Measure accuracy for all labels.
            Measure Matthews Correlation Coefficient for all labels.
        end for
    end for
end for
```

classifier and decision tree classifier. Having separate lists for the 3 datasets, 4 data transformations (including no transformation) and 3 algorithms, I generated one set of classification results for each combination of these lists and evaluated the results. As a result, 36 sets of clustering results were generated, and for each 2 evaluation metrics were measured: accuracy and Matthew's correlation coefficient.

In this thesis, I have used discriminative and discrete classifiers since the labels for patients in my metaproteomic abundances were discrete labels. However, simultaneously I wanted to check whether parametric and non-parametric classifiers would exhibit differences in classifying eigen representation of metaproteomic abundance. I have used accuracy and MCC to evaluate the quality of predictions by the classifiers implemented on the eigendecomposition of the metaproteomic abundance data to check prediction quality for all labels. Additionally, I checked the precision and recall for the label ”control”, since these metrics provide improved interpretation of a single label, whereas, accuracy and MCC provide better interpretation for all (several) labels.

The four data transformations that were observed in this experiment were between original (no transformation), normalized, eigendecomposition (k largest eigenvectors) and eigendecomposition (2k largest eigenvectors), where k denotes the k-largest eigenvectors used from self-tuning eigendecompositions. An overview of all the clustering and classification algorithms used in both experiments is shown in table 4.2 with their corresponding python package that was used. The eigendecomposition used in this case for classification tasks is the same eigendecomposition used in the self-tuning algorithm as described by Sun [2020]. Eigen (k) and (2k) refers to k and 2k largest eigenvectors selected for analysis, where k denotes the number of class labels for the given data (7 for dataset 1, 3 for dataset 2 and 9 for dataset 3).

Experiment 2 helped to understand the efficiency of using eigendecomposition techniques, against no transformation (original) and normalization, as a pre-processing step prior to predictive analytics of patients with metaprotein abundance data. It was also verified whether increasing the number of k for k largest eigenvectors, had any major influence on improving the performance of the classifier algorithms.

4.5 Hardware and Software

Hardware: All the experiments were implemented and executed on a personal laptop with an Intel core i3 (7th generation) processor, 8 gigabytes of random access memory (RAM) and 256 gigabytes of solid state drive.

Software: All the experiments were coded and implemented in Python version 3.10. The codes were written and compiled inside a jupyter environment. Python packages were used for the data pre-processing, transformation and experiments.

In table 4.2, I have listed all the python packages used for all the experiments along with their corresponding package version to ensure reproducibility.

Table 4.2: Overview of python packages and their versions used in the experiments.

	Task	Python package	Package version
normalization	transformation	sklearn.preprocessing. MinMaxScaler()	scikit-learn 1.2.2
principal component analysis (PCA)	transformation	sklearn.decomposition. PCA(n_components = 2)	scikit-learn 1.2.2
k-means	clustering	sklearn.cluster. KMeans(n_clusters = 3)	scikit-learn 1.2.2
agglomerative	clustering	sklearn.cluster. AgglomerativeClustering (n_clusters = 3)	scikit-learn 1.2.2
spectral (python package)	clustering	sklearn.cluster. SpectralClustering (n_clusters = 3)	scikit-learn 1.2.2
spectral (NJW)	clustering	NJW eigen transform + k-means	scikit-learn 1.2.2 for k-means
spectral (self-tuning)	clustering	self-tuning eigen transform + k-means	scikit-learn 1.2.2 for k-means
nearest centroid	classification	sklearn.neighbors. NearestCentroid()	scikit-learn 1.2.2
k-nearest neighbour	classification	sklearn.neighbors. KNeighborsClassifier (n_neighbors = 5)	scikit-learn 1.2.2
decision tree	classification	sklearn.tree. cDecisionTreeClassifier (criterion = ”gini”)	scikit-learn 1.2.2

5. Evaluation

In this chapter, I have evaluated the results of two experiments (see chapter 4), on 3 different metaprotein datasets (IBD dataset, NASH dataset and combined dataset) and several transformation techniques. In section 5.1, results for experiment 1 is depicted through bar plots of each execution, and box plots and mean values summarising the results. The same have been depicted for experiment 2 in section 5.2. Afterwards, in section 5.3, I have compared the datasets, especially dataset 3 against datasets 1 and 2. Finally in section 5.4, I have answered the research questions as proposed in chapter 1.

5.1 Experiment 1: Comparison of Clustering

In the first experiment, I investigated the influence of different transformations and algorithms on the clustering of other metaproteomic datasets (Research Question 1 as in section 1.3). Furthermore, I analyzed whether combining two metaproteomic datasets into a more extensive dataset might improve the performance of clustering and classification. Therefore, I evaluated the resulting clusters on the silhouette coefficient and adjusted rand index (see chapter 4). The following comparisons are investigated in experiment 1:

- Normalization, and PCA as transformation technqiues prior to clustering in comparison to no prior transformation (original dataset).
- Spectral clustering techniques against hierarchical clustering technique and variations within spectral clustering techniques (different similarity graphs).
- Combining several metaproteomic abundance datasets to form a single dataset.

Dataset 1 (Inflammatory Bowel Diseases) - The silhouette coefficient and adjusted rand index for each clustering result are displayed in bar plots in figure 5.1. Spectral clustering (NJW) provided the highest silhouette coefficient for all the transformations, with the original (no transformation) outperforming other transformations.

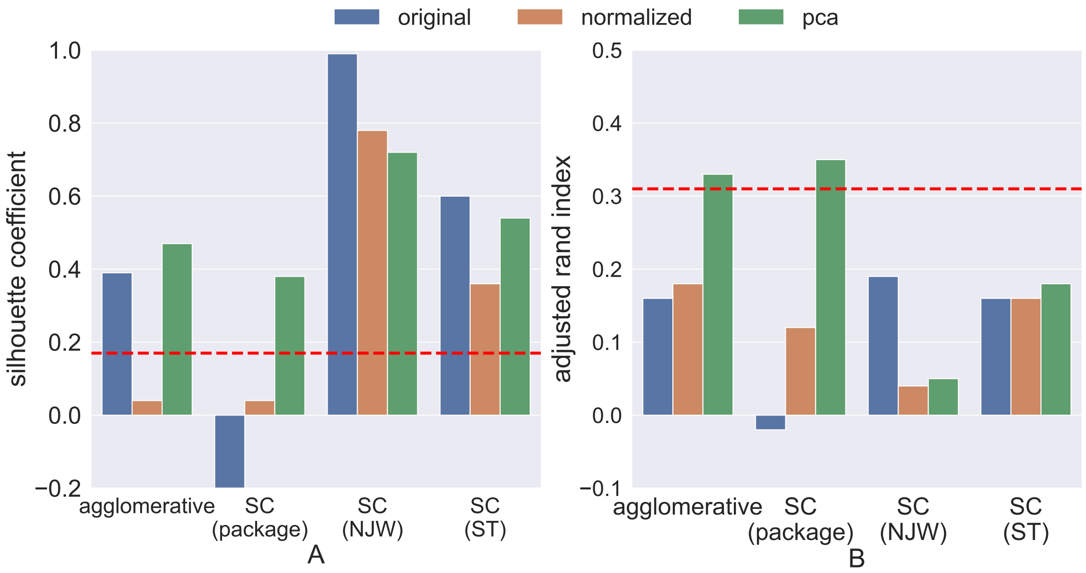


Figure 5.1: Bar plots comparing A) silhouette coefficient and B) adjusted rand index for Dataset 1 subdivided into individual data transformations. Red dashed lines represent the benchmark silhouette coefficient (0.17) and adjusted rand index (0.31). SC = Spectral Clustering, NJW = Ng-Jordan-Weiss algorithm and ST = Self-Tuning algorithm.

PCA transformation was effective for dataset 1, often outperforming normalization, especially for adjusted rand index values. Benchmark silhouette coefficient (0.17) was crossed by most except original and normalized transformation in agglomerative and spectral clustering (package) of k-nn graph. Benchmark adjusted rand value (0.31) was crossed only by PCA for agglomerative and spectral clustering (package).

Dataset 2 (Non-Alcoholic Fatty Liver Disease) - The silhouette coefficient and adjusted rand index for each clustering result are displayed in bar plots in figure 5.2. Spectral clustering (NJW) provided the highest silhouette score for all the transformations, with original (no transformation) outperforming other transformations. PCA outperformed normalization in terms of silhouette coefficient, however, in terms of adjusted rand index, normalization outperformed PCA for agglomerative and spectral clustering (package). Benchmark silhouette coefficient (0.02) was crossed by most except the original for spectral clustering (package). Benchmark adjusted rand index (0.44) could not be crossed in any case. This could be due to the fact that in the original research, Bray-Curtis dissimilarity was used as a proximity measure between data points, which might be more efficient in portraying the underlying structure of metaproteomic data, in the context of the class labels.

Dataset 3 (IBD and NAFLD) - The silhouette coefficient and adjusted rand index for each clustering result are displayed in bar plots in figure 5.3. Spectral clustering (NJW) provided the highest silhouette score for all the transformations, with normalization slightly outperforming other transformations in this case. However, for other algorithms, PCA mostly outperformed normalization in terms of silhouette scores as well as in terms of adjusted rand index. Benchmark values are not available for this dataset.

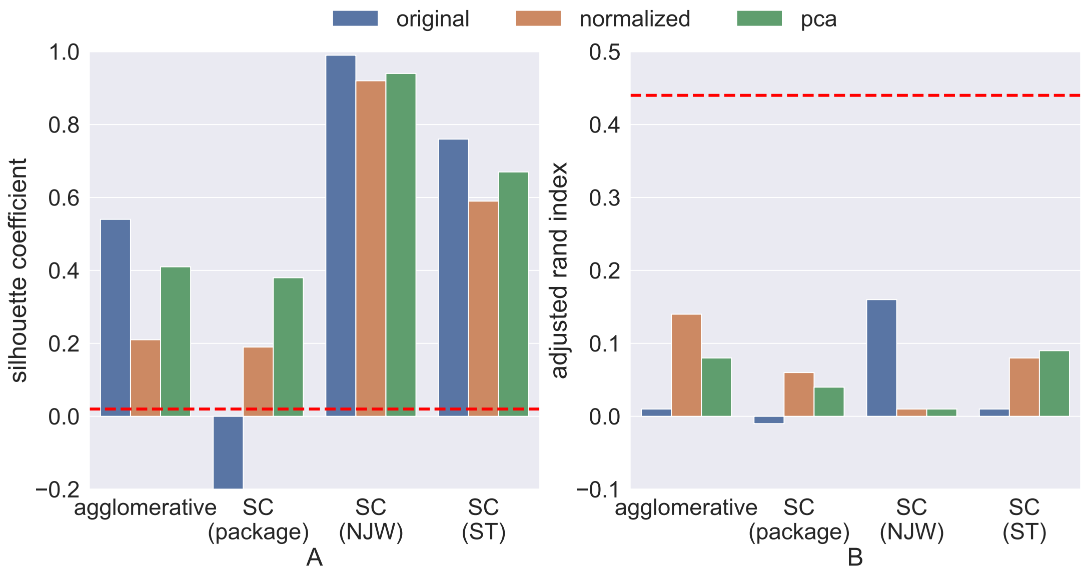


Figure 5.2: Bar plots comparing A) silhouette coefficient and B) adjusted rand index for Dataset 2 subdivided into individual data transformations. Red dashed lines represent the benchmark silhouette coefficient (0.02) and adjusted rand index (0.44). SC = Spectral Clustering, NJW = Ng-Jordan-Weiss algorithm and ST = Self-Tuning algorithm.

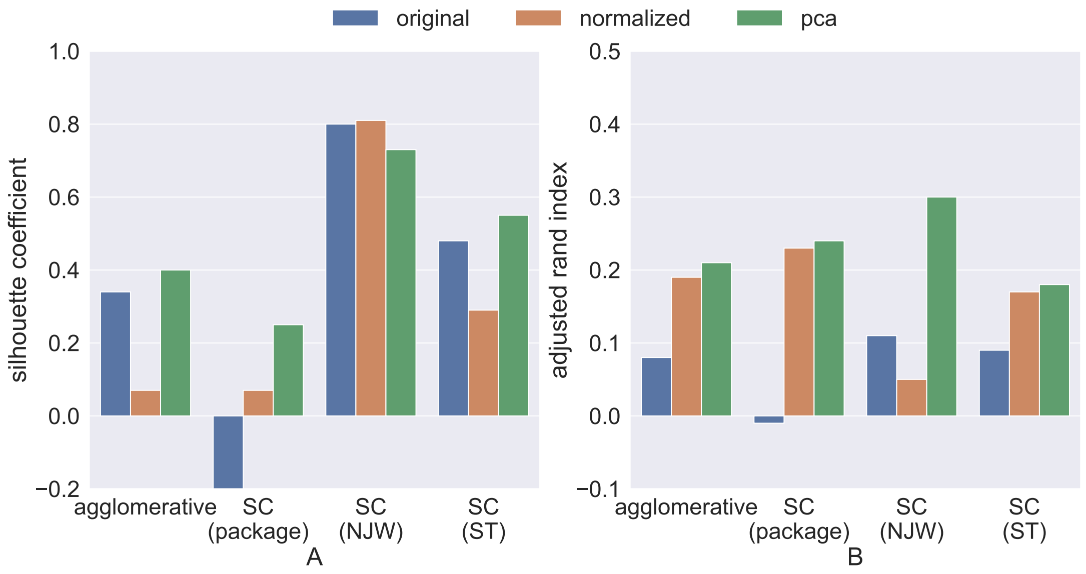


Figure 5.3: Bar plots comparing A) silhouette coefficient and B) adjusted rand index for Dataset 3 subdivided into individual data transformations. No benchmark values are available. SC = Spectral Clustering, NJW = Ng-Jordan-Weiss algorithm and ST = Self-Tuning algorithm.

Mean values of silhouette coefficient and adjusted rand index are presented in table 5.1 and average distributions have been presented in box plots in figure 5.4. In terms of data transformation, the original provided the best silhouette score for the NJW algorithm. PCA provided the highest silhouette scores on several occasions due to the fact that PCA reduces the number of dimensions (2) and a lower dimensional representation often results in a better silhouette score due to a reduction in noise and redundancy [Karamizadeh et al., 2013]. Furthermore, considering the adjusted rand index, PCA mostly outperformed the original data. Spectral clustering already reduces the dimensionality of the data, i.e., a PCA is not necessary as a prior transformation step. That is why no transformation performs better than PCA in figure 5.1, 5.2 and 5.3. As a result, PCA was not observed for experiment 2.

In terms of comparing clustering algorithms, spectral clustering algorithms (NJW and ST) outperforms hierarchical clustering in terms of cluster separation (silhouette coefficient) with a fully-connected graph. However, for label separation (adjusted rand index), agglomerative clustering performed similarly to spectral clustering. When compared to benchmark values, both hierarchical and spectral clustering as a whole, has provided improved performance in terms of silhouette coefficient, depicting superior cluster separation. However, benchmark values were not met when compared in terms of the adjusted rand index, especially for dataset 2. One reason for this could be that the original research for dataset 2 used Canberra distance as a distance measure, compared to the Euclidean distance used in this research.

Table 5.1: Mean silhouette and adjusted rand coefficients for all categories: datasets, data transformations and clustering algorithms.

	datasets			**data transformations**			**clustering algorithms**			
	d1	**d2**	**d3**	**original**	**nomr-arlized**	**pca**	**agglo-merative**	**spectral (package)**	**spectral (NJW)**	**spectral (ST)**
silhouette	0.42	0.53	0.36	0.41	0.36	0.54	0.31	0.04	0.85	0.54
adjusted rand	0.16	0.06	0.15	0.08	0.12	0.17	0.15	0.11	0.10	0.12

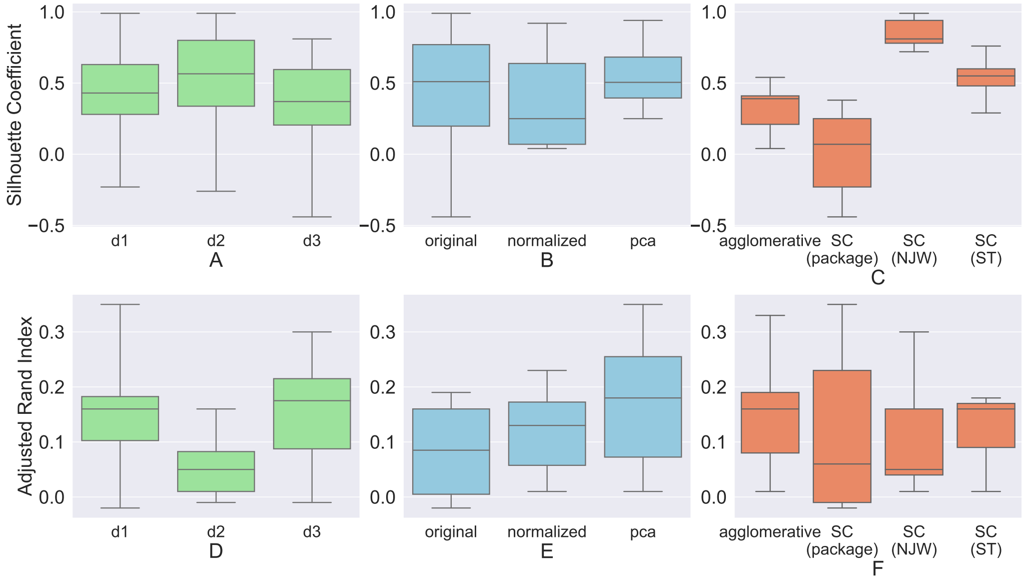


Figure 5.4: Box plots comparing the accuracy and MCC for datasets(A,D), data transformations (B,E) and classification algorithms (C,F).

Dataset 3 did not provide significant improvement for silhouette coefficients over datasets 1 and 2. However, in terms of the adjusted rand index, it showed consistent improvement. To summarize, using spectral clustering (NJW) without transformation showed the best performance in terms of silhouette coefficient and adjusted rand index on metaproteomic abundance data. It could be argued that PCA is a useful step when applying spectral clustering algorithms, however, comes with the redundancy of two eigendecompositions to be applied.

To summarise, the following has been observed in experiment 1:

- The performance of the algorithms did not vary significantly in most cases, for different algorithms, with PCA providing consistently good results. Original (no transformation) performed better for spectral clustering (NJW).
- Spectral clustering with fully-connected graphs (NJW and self-tuning algorithms), provides optimal performance for clustering with silhouette values reaching up to an average of 0.85 (NJW) and 0.54 (self-tuning).
- Dataset 3 did not provide improved results for the silhouette coefficient, however, significant improvement was observed in some cases, for the adjusted rand index, compared to datasets 1 and 2.

5.2 Experiment 2: Comparison of Classification

In the second experiment, I investigated the influence of different transformations and classifiers for predicting patient disease from metaproteomic abundance datasets. Furthermore, I analyzed whether combining two metaproteomic datasets into a more extensive dataset might improve the performance of classification. Therefore, I evaluated the resulting predictions on the accuracy and Matthew's Correlation Coefficient (mcc) (see chapter 4). Furthermore, 5-fold cross-validation was performed on dataset 2 for 10,000 iterations (figure 5.7), for appropriate comparison against the source research, which implemented k-fold cross-validation as well for k = 5. The following comparisons are investigated in experiment 1:

- Data transformation techniques - original, normalized, eigen (k) and eigen (2k), prior to classification.
- Classifiers: nearest centroid, k-nearest neighbour and decision tree.
- Datasets especially dataset 3, and realize improvement of combining 1 and 2.

Dataset 1 (Inflammatory Bowel Diseases) - The accuracy and MCC values for each clustering result are displayed in bar plots in figure 5.5. Eigen (k) transformation consistently provided better accuracy (0.47) as well as mcc (0.37), especially for the nearest centroid and decision tree classifier (same values for both). However, k-nn classifiers provided better performance with normalized data. Eigen (2k) did not provide any improvement over eigen (k) for this dataset.

Dataset 2 (Non-Alcoholic Fatty Liver Diseases) - The accuracy and Matthew's correlation coefficient (mcc) values for each clustering result are displayed in bar plots in figure 5.6. Eigen (k) transformation on a decision tree classifier provided the

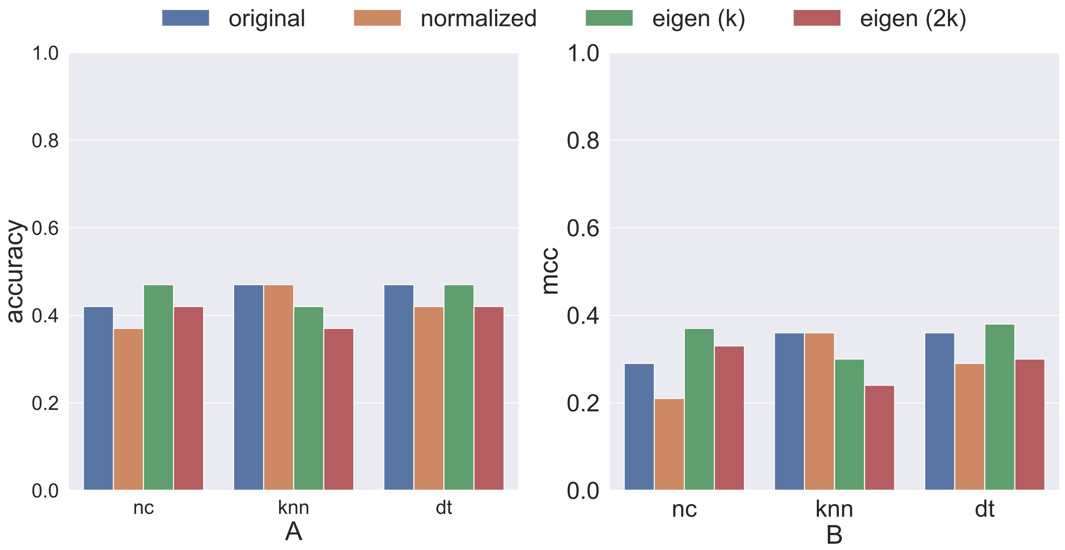


Figure 5.5: Bar plots comparing A) accuracy and B) Matthew's correlation coefficient (mcc) for Dataset 1 subdivided into individual data transformations. No benchmark values are available. nc = nearest centroid, knn = k-nearest neighbours and dt = decision tree algorithms.

best accuracy (0.55). Eigen (2k) provided significant improvement over eigen (k) for the nearest centroid classifier.

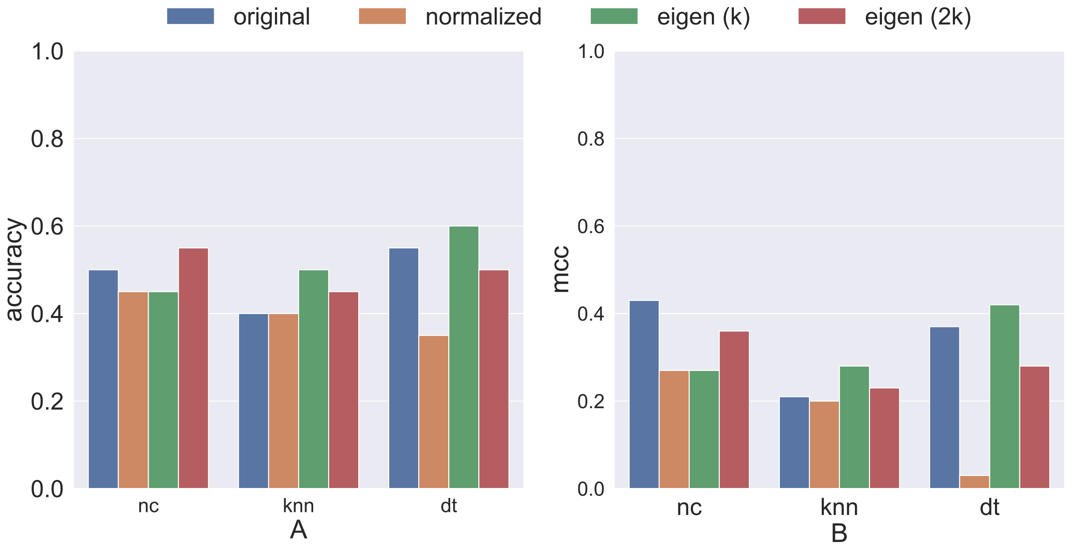


Figure 5.6: Bar plots comparing A) accuracy and B) Matthew's correlation coefficient (mcc) for Dataset 2 subdivided into individual data transformations. Benchmark values for dataset 2 are compared in figure 5.7. nc = nearest centroid, knn = k-nearest neighbours and dt = decision tree algorithms.

For benchmark comparison, with 5-fold cross-validation, bar plots are plotted in figure 5.7. None of the classifiers could cross the benchmark accuracy of 0.86. This could be due to the fact that the original research [Sydor et al., 2022] used Canberra

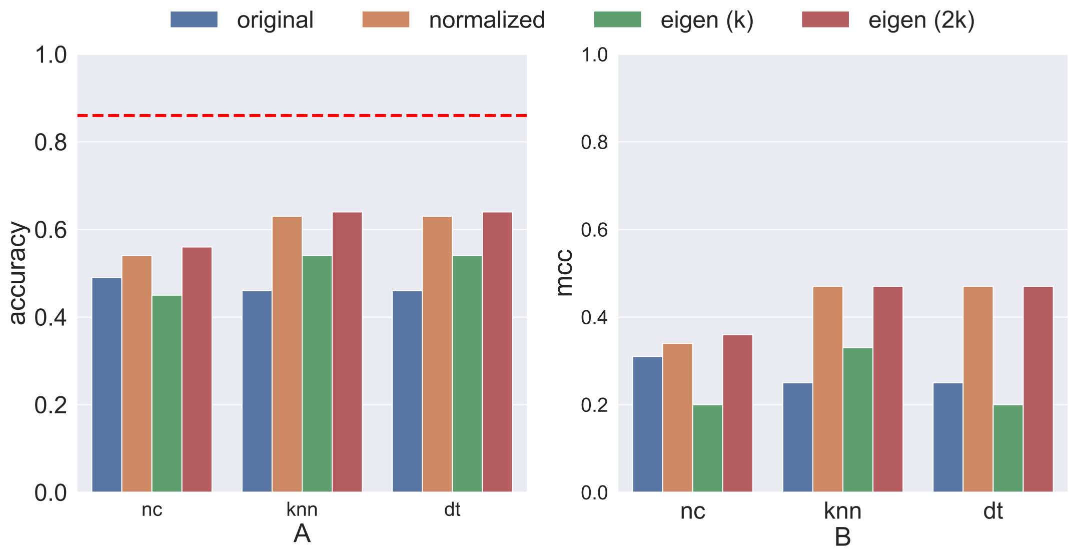


Figure 5.7: Bar plots comparing A) accuracy and B) Matthew's correlation coefficient (mcc) for Dataset 2 subdivided into individual data transformations. Red dashed lines represent the benchmark accuracy (0.86). No benchmark mcc is available. nc = nearest centroid, knn = k-nearest neighbours and dt = decision tree algorithms.

distance to calculate proximity between data points which might have contributed to signifying the underlying structure of the class labels. However, in this case for dataset 2, eigen (2k) provided consistently better accuracy (highest 0.64 for both k-nn and decision tree) and MCC (highest 0.47 for both accuracy and mcc) over other transformations.

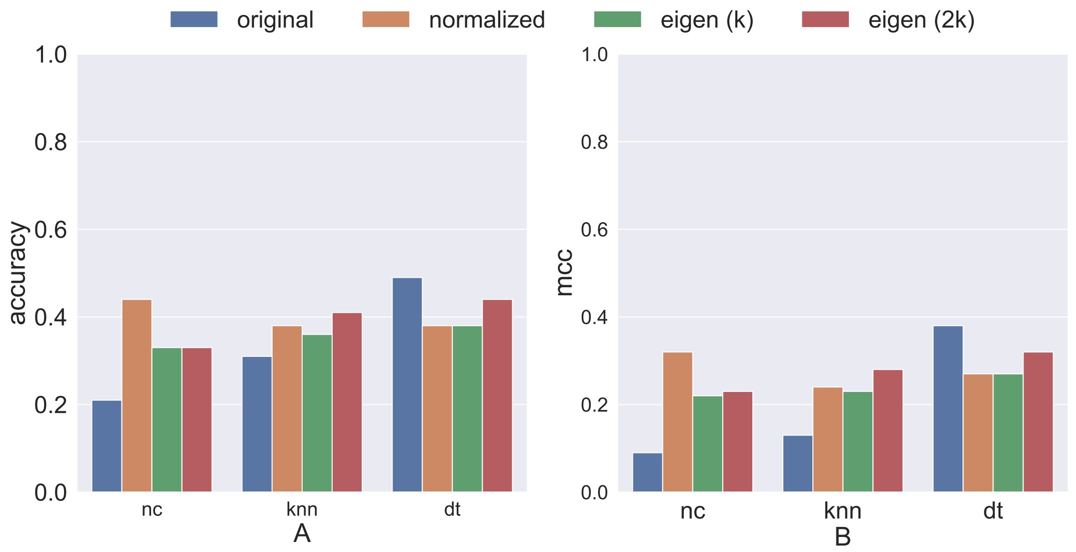


Figure 5.8: Bar plots comparing A) accuracy and B) Matthew's correlation coefficient (mcc) for Dataset 3 subdivided into individual data transformations. Red dashed line represents the benchmark accuracy (0.86). No benchmark mcc is available. nc = nearest centroid, knn = k-nearest neighbours and dt = decision tree algorithms..

Dataset 3 - The accuracy and Matthew's correlation coefficient (mcc) values for each clustering result are displayed in bar plots in figure 5.8. Original (no transformation) on the decision tree classifier, provided the best accuracy (0.49) as well as mcc (0.38). Normalization provided the second-best results for the nearest centroid classifers (accuracy - 0.44 and MCC - 0.32). And eigen (2k) provided the best performance for k-nn classifier (accuracy - 0.41 and MCC - 0.28).

In terms of data transformation, both eigendecompositions (eigen (k) and (2k)) proved to be highly efficient for classification tasks. However, when means and medians are compared, eigendecompositions and normalization perform equally. Additionally, choosing 2k largest eigenvectors during eigendecomposition as a precursor to classification significantly improves accuracy and MCC over k largest eigenvectors.

Table 5.2: Mean accuracy and Matthew's correlation coefficient (mcc) for all categories: datasets, data transformations and clustering algorithms

	datasets			data transformations				classifier		
	d1	d2	d3	original	norm-arlized	eigen (k)	eigen (2k)	nearest centroid	k-nn	decision tree
acuuracy	0.43	0.48	0.37	0.44	0.46	0.46	0.48	0.44	0.45	0.48
MCC	0.32	0.28	0.25	0.29	0.29	0.29	0.32	0.28	0.29	0.32

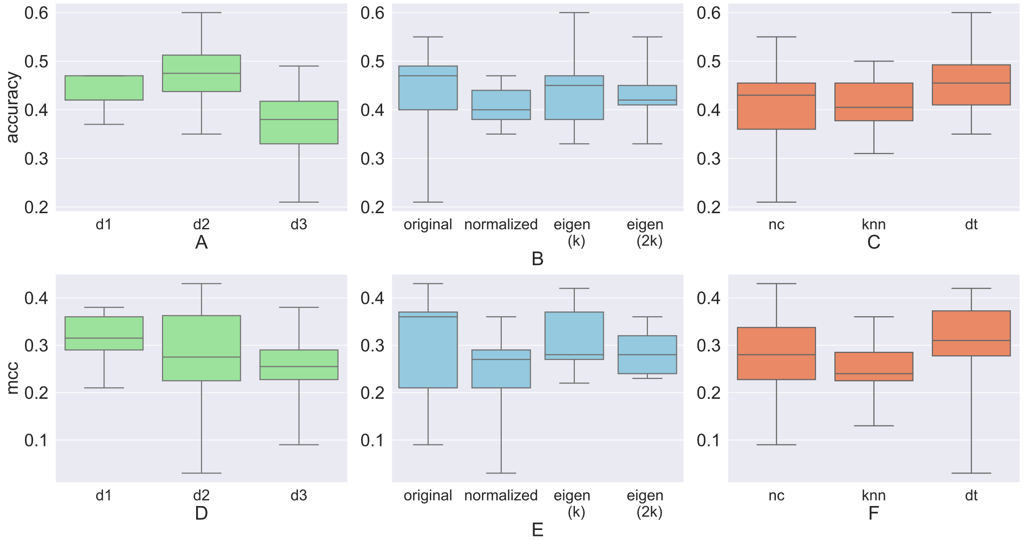


Figure 5.9: Box plots comparing the accuracy and MCC for datasets(A,D), data transformations (B,E) and classification algorithms (C,F).

In terms of the performance of classification algorithms, all 3 algorithms - nearest centroid, k-nn and decision tree, provided similar performance, both in terms of accuracy and MCC. In fact, the decision tree performed slightly better with the highest mean MCC of 0.44.

Comparing the datasets, dataset 3 could not outperform their source datasets 1 and 2. This could be potentially due to the larger number of labels present with a high class imbalance. However, it could be possible that control patients could be better analysed by combining metaproteins datasets. To investigate this further, I have plotted box plots (figure 5.10) to compare the precision and recall for the label ”control” over all 3 datasets in section 5.3

To summarise, the following has been observed in experiment 2:

- Eigen (k) decompositions performed consistently well for datasets 1 and 2. Additionally, eigen (2k) provided a significant improvement over eigen (k) in the benchmark comparison of dataset 2.
- All classification algorithms performed quite similarly with the decision tree providing slightly better results with a mean accuracy of 0.48 and mean mcc of 0.32.
- Dataset 3 (mean accuracy - 0.37 and mean mcc - 0.25) did not provide any improvement. However, it showed almost similar performance to dataset 1 (mean accuracy - 0.43 and mean mcc - 0.32).

5.3 Comparison of datasets

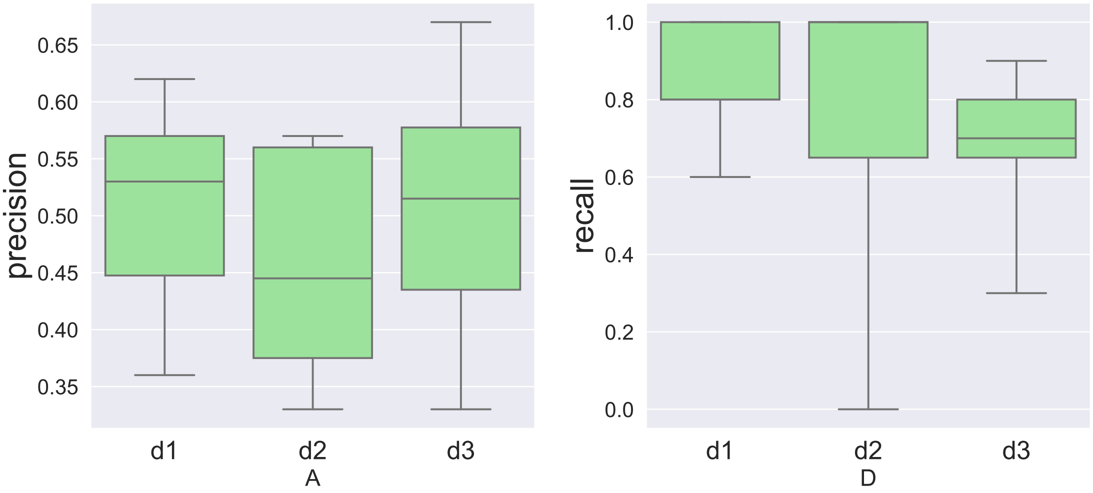


Figure 5.10: Box plots comparing the A) precision and B) recall. for the 3 datasets.

Dataset 3 provided a similar performance to the other datasets only during clustering for adjusted rand index. It had lower silhouette coefficients for clustering, otherwise. Additionally, it did not provide improvement of accuracy and mcc as well, in comparison to datasets 1 and 2. To check, whether it could at least provide improved precision and recall for the label ”control” out of the 9 class labels that it contains, I have compared them in figure 5.10.

In terms of precision, it provided higher mean precision than dataset 2 and a higher precision was achieved over dataset 1 however slightly less than dataset 1. This shows that a deeper investigation into how metaproteomic abundances could be merged could have promising results on how to better distinguish between sick and controlled patients. It could be possible that the reason why dataset 3 could not outperform other datasets is the same reason why dataset 2 could not cross benchmark values (apart from class imbalance): the selection of a better proximity measure. Euclidean distance performs better for data that has a low number of normally distributed features. However, metaproteomic abundance data is high-dimensional and is often not normally distributed.

5.4 Discussion of Research Questions

In the following, I will answer the research questions as proposed in section 1.3.

Research Question 1: How well does spectral clustering group patients from metaproteomic abundances, in terms of internal and external validation indices, in comparison to hierarchical clustering?

Answer: Spectral clustering (NJW algorithm) provided sufficiently high performance, in terms of clustering metaproteomic abundance data. It has outperformed hierarchical clustering (agglomerative) by an average silhouette coefficient of 0.54 (0.85 (NJW) and 0.31 (agglomerative)). In terms of the adjusted rand index, it performed quite similarly to the hierarchical (average of 0.15), with a slightly lower average value of 0.10. Spectral clustering (self-tuning), similarly outperformed hierarchical in terms of the silhouette with an average increase of 0.23, and almost similar in terms of adjusted rand index with an average of 0.03 lower than hierarchical. Spectral clustering with k-nn graph (package) could not outperform hierarchical clustering, for k = 5. Overall, it can be stated that spectral clustering algorithms, based on fully-connected graphs can have improved cluster separation, without considering class labels, and almost similar cluster separation considering class labels than hierarchical clustering.

Research Question 2: To what extent do data transformation techniques such as normalization and PCA improve clustering performance for metaproteomic abundances, in terms of internal and external validation indices?

Answer: PCA has outperformed normalization, in terms of both silhouette coefficient and adjusted rand index, making it a suitable transformation technique to be applied prior to clustering, especially agglomerative clustering, of metaproteomic abundance data. It provided an average increase of 0.13 over the original data, in terms of silhouette and an average increase of 0.09 in terms of adjusted rand index. However, no transformation provided the best silhouette coefficient (0.97) in the best-case scenario i.e. for spectral clustering (NJW algorithm) in most cases (negligibly lower than normalization for dataset 3). Normalization, while providing an average increase of 0.04 over original data in terms of adjusted rand index, provided an average decrease of 0.05 in terms of silhouette. As a result, PCA is a very useful data transformation technique to consider while clustering metaproteomic abundance data.

Research Question 3: How much better accuracy and Matthews Correlation Coefficient could be achieved over normalization, if eigendecomposition was applied as a data transformation step, to predict patient labels from metaproteomic abundances?

Answer: On average, eigendecomposition performed quite similarly when compared to normalization, with eigen (2k) providing slightly better accuracy (average increase of 0.02) and MCC (average increase of 0.03) over eigen (k). However, this depends on various characteristics of the dataset. For example, fewer class labels in a dataset can lead to better performance (dataset 2). However, further investigation can provide deeper insights how eigendecomposition could be optimised to provide even better performance while handling a high number of class labels in the dataset.

Research Question 4: How much improvement could be achieved, in terms of clustering and classification, if two metaproteomic abundances datasets were combined into one?

Answer: No improvement of predictions was achieved either in terms of clustering and classification when two metaproteomic abundance datasets were combined. However, in the case of clustering, it could achieve a similar average adjusted rand index score (0.15 for dataset 3 compared to 0.16 for dataset 1 and 0.06 for dataset 2), depicting the need for further investigation in this arena. As a huge proportion of the metaproteins could be found in samples from the same environment, e.g., human gut, it would be quite useful to formulate a method to combine several metaproteomic abundance datasets into one, for improved clustering and/or classification.

6. Conclusion

The aim of this study was to investigate the impact of spectral clustering on metaproteomic abundance data and to classify these data based on eigendecomposition. In this section, we will discuss the implications of our findings and their significance in the broader context of metaproteomics research.

Spectral clustering, widely popular in the domain of image segmentation, can be efficient in grouping data points for numerical features. The ability to highlight gaps between eigenvalues serves to make spectral clustering an efficient tool for various tasks, e.g., image segmentation and metaproteomic abundance data. With a growing number of research on omics data, especially with modern machine learning algorithms, we can expect the following in the future:

- Improved protein identification.
- Increased understanding of microbial communities.
- Identification of several biomarkers of diseases.
- Development of new targeted therapies.

Old methods are often rediscovered with new capabilities and this thesis aimed to achieve that for eigendecomposition. There have been a handful of research on the application of spectral clustering on metaproteomic abundance data and the results show it deserved a deeper investigation, to further improve its capabilities on very large datasets to find similar data points.

This study also highlighted the importance of an appropriate data transformation techniques and clustering algorithms for metaproteomic abundance data. It was found that no transformation and PCA, combined with spectral clustering are possible effective techniques for grouping patients, particularly when used with fully-connected graphs. This is because spectral clustering can identify clusters with high intra-cluster similarity and low inter-cluster similarity. It can sufficiently outperform hierarchical clustering, which is still a popular choice in metaproteomic research.

It was also found that different datasets have significantly different impacts on clustering and classification performance, with an inverse proportionality observed against the number of class labels present in the data. Although dataset 3 did not sufficiently improve results, its formation could still be an essential method that could be utilized to benchmark future studies on metaproteomic abundance data. The combination process could be further improved and could prove useful in a binary classification of diseased against control patients.

Focusing on classification, eigen (k) and eigen (2k) transformations proved to be useful transformation techniques to improve accuracy and MCC. However, the choice of classification algorithm needs to be considered for optimal output. Classifiers providing a linear decision boundary, such as the nearest centroid classifier, seem to perform better with eigendecomposition. k-nn and decision trees provided better performance for both normalization and eigendecomposition confirming that the label structure of metaproteomic abundances could be better sorted by non-linear decision boundaries on the class label. However, on average, all classifiers performed quite similarly, depicting the ease of analysis of eigenvalues and vectors by any given algorithm.

In conclusion, the study provides important insights into the impact of spectral clustering on metaproteomic abundance data and its classification based on eigendecomposition. I recommend using normalization as a data transformation technique and spectral clustering, particularly with fully-connected graphs, for clustering metaproteomic abundance data, as well as classification of eigen-transformed data.

6.1 Scope of Future Work

In retrospect of the key findings of this research, I recommend investigating the following research areas, for further improvements in the clustering and classification of metaproteomic abundance data:

1. **Impact of applying other spectral clustering algorithms:** There exists more than 15 types of spectral clustering algorithms, 2 of which (Ng-Jordan-Weiss and Self-tuning) were investigated in this research. It would be interesting to see if several other spectral clustering algorithms would provide similar improvements for clustering tasks.
2. **Impact of different proximity measures on the performance of spectral clustering** - Instead of using Euclidean distance as part of the equation to calculate pairwise similarity/adjacency, other proximity measures could be investigated, e.g., Bray-Curtis dissimilarity, cosine similarity.
3. **Binary classification after combining several metaproteomic abundance datasets** - Several types of diseased patients could all be labelled the same as ”sick” and the rest as ”control” and predictive analytics could be performed on the combined dataset to identify biomarkers for healthy patients.
4. **Impact of different other classifiers to predict patients from metaproteomic abundance datasets:** Apart from using nearest centroid, k-nn and decision tree classifiers, the impact of using other advanced classifiers such as XGBoost, Random Forest and Neural Networks could be investigated.

5. **Choice of k for k-largest eigenvectors** - While in this research, in the context of choosing the number of largest eigenvectors to analyse, I have only investigated between eigen (k) and eigen (2k). It could be further investigated whether even a larger number of k could potentially be useful to deal with the high-dimensionality of metaproteomic abundance datasets.
6. **Impact of ϵ-neighbourhood graphs** - Most spectral clustering algorithms either use a k-nn graph or a fully-connected graph. However, a new algorithm could be generated to use a ϵ-neighbourhood graph (see section 2.2.1). This can help to reduce the complexity in the similarity matrix of the data, since a similarity value below a certain threshold, would be considered as 0 and would not affect the calculations for ϵ-neighbourhood graph. The impact of different ϵ values, in this case, could also be investigated.

In this regard, as an extension of this work, I have already compiled 15 spectral clustering algorithms (including Ng-Jordan-Weiss and Self-tuning) and compared them in detail for the purpose of publishing an exploratory survey paper on spectral clustering algorithms. This would help researchers in any domain to be able to know and compare all spectral clustering algorithms along with their specialized applications, and to be used for various tasks and research. Furthermore, I am also investigating whether clustering can be performed, generically for any data, on a Microsoft Excel file. The possibility of this extends to the formulation of spectral clustering algorithms, as well, on an Excel file. This would largely diminish the need for understanding a programming language e.g., python, to be able to implement clustering or spectral clustering algorithm, in addition to the steps of the algorithms being more explainable in this format.

Bibliography

Charu C. Aggarwal and Haixun Wang. *A Survey of Clustering Algorithms for Graph Data*, pages 275–301. Springer US, Boston, MA, 2010. ISBN 978-1-4419-6045-0. doi: 10.1007/978-1-4419-6045-0_9. (cited on Page ix and 16)

Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9 (8):1295, Aug 2020. ISSN 2079-9292. doi: 10.3390/electronics9081295. URL http://dx.doi.org/10.3390/electronics9081295. (cited on Page 16)

Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020. (cited on Page xiii and 23)

Renzo Angles and Claudio Gutierrez. Survey of graph database models. *ACM Computing Surveys*, 40(1):2, 2008. ISSN 0360-0300. URL https://doi.org/10.1145/1322432.1322433. (cited on Page ix and 11)

Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2): 49–60, Jun 1999. doi: 10.1145/304181.304187. (cited on Page 16)

Geoffrey H Ball and David J Hall. Isodata, a novel method of data analysis and pattern classification. Technical report, Stanford research inst Menlo Park CA, 1965. (cited on Page 14)

James C. Bezdek. Cluster validity with fuzzy sets. *Journal of Cybernetics*, 3(3): 58–72, 1973. doi: 10.1080/01969727308546047. (cited on Page 21)

James C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, 1987. (cited on Page 21)

Marcel Brun, Chao Sima, Jianping Hua, James Lowey, Brent Carroll, Edward Suh, and Edward R. Dougherty. Model-based evaluation of clustering validation measures. *Pattern Recognition*, 40(3):807–824, 2007. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2006.06.026. URL https://www.sciencedirect.com/science/article/pii/S0031320306003104. (cited on Page xiii, 15, and 21)

T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3(1):1–27, 1974. doi: 10.1080/03610927408827101. (cited on Page 21)

M Emre Celebi. *Partitional clustering algorithms*. Springer, 2014. URL https://link.springer.com/content/pdf/10.1007/978-3-319-09259-1.pdf. (cited on Page xiii and 16)

Kai Cheng, Zhibin Ning, Xu Zhang, Leyuan Li, Bo Liao, Janice Mayne, Alain Stintzi, and Daniel Figeys. Metalab: an automated pipeline for metaproteomic data analysis. *Microbiome*, 5:1–10, 2017. (cited on Page 7)

Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13, 2020. (cited on Page 26)

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967. (cited on Page 24)

Kellen G Cresswell, John C Stansfield, and Mikhail G Dozmorov. Spectraltad: an r package for defining a hierarchy of topologically associated domains using spectral clustering. *BMC bioinformatics*, 21:1–19, 2020. (cited on Page 27 and 30)

David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909. (cited on Page 20 and 21)

R López De Mántaras. A distance-based attribute selection measure for decision tree induction. *Machine learning*, 6(1):81–92, 1991. (cited on Page 21)

J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973. doi: 10.1080/01969727308546046. (cited on Page 21)

Absalom E. Ezugwu, Amit K. Shukla, Moyinoluwa B. Agbaje, Olaide N. Oyelade, Adán José-García, and Jeffery O. Agushaka. Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature. *Neural Computing and Applications*, 33(11):6247–6306, Jun 2021. ISSN 1433-3058. URL https://doi.org/10.1007/s00521-020-05395-4. (cited on Page ix and 16)

James S. Farris. On the cophenetic correlation coefficient. *Systematic Zoology*, 18(3):279–285, Sep 1969. doi: 10.2307/2412324. (cited on Page 21)

Maurizio Filippone, Francesco Camastra, Francesco Masulli, and Stefano Rovetta. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190, 2008. (cited on Page 10)

Edward W Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *biometrics*, 21:768–769, 1965. (cited on Page 23)

Kazuhisa Fujita. Approximate spectral clustering using both reference vectors and topology of the network generated by growing neural gas. *PeerJ Computer Science*, 7, 2021. doi: 10.7717/peerj-cs.679. (cited on Page 16)

Y. Fukuyama and M. Sugeno. A new method of choosing the number of clusters for the fuzzy c-mean method. In *Proc. 5th Fuzzy Syst. Symp., 1989*, pages 247–250, 1989. (cited on Page 21)

Guojun Gan, Chaoqun Ma, and Jianhong Wu. Data clustering: Theory, algorithms, and applications. page 183–298, 2007. doi: 10.1137/1.9780898718348. (cited on Page xiii and 21)

Garima, Hina Gulati, and P.K. Singh. Clustering techniques in data mining: A comparison. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 410–415, 2015. (cited on Page xiii and 16)

Jeovanis Gil. *Contribución al estudio proteómico de la acetilación en residuos de lisina y del papel de la desacetilasa SIRT1 en células humanas.* PhD thesis, 12 2017. (cited on Page ix and 6)

John C Gower and Gavin JS Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 18 (1):54–64, 1969. (cited on Page 15)

Simon Günter and Horst Bunke. Validation indices for graph clustering. *Pattern Recognition Letters*, 24(8):1107–1113, 2003. doi: 10.1016/s0167-8655(02)00257-x. (cited on Page 19 and 21)

Lars Hagen and Andrew B Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems*, 11(9):1074–1085, 1992. (cited on Page 18)

M. Halkidi and M. Vazirgiannis. Clustering validity assessment: Finding the optimal partitioning of a data set. *Proceedings 2001 IEEE International Conference on Data Mining*, page 187–194, 2001. doi: 10.1109/icdm.2001.989517. (cited on Page 20 and 21)

M. Halkidi, M. Vazirgiannis, and Y. Batistakis. Quality scheme assessment in the clustering process. *Principles of Data Mining and Knowledge Discovery*, page 265–276, 2000. doi: 10.1007/3-540-45372-5_26. (cited on Page 21)

Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2/3):107–145, Oct 2001. doi: 10.1023/a:1012801612483. (cited on Page 21)

Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Clustering validity checking methods. *ACM SIGMOD Record*, 31(3):19–27, 2002. doi: 10.1145/601858.601862. (cited on Page 21)

Pauline Hardouin, Raphael Chiron, Hélène Marchandin, Jean Armengaud, and Lucia Grenga. Metaproteomics to decipher cf host-microbiota interactions: overview, challenges and future perspectives. *Genes*, 12(6):892, 2021. (cited on Page ix and 1)

Pei Jihong and Yang Xuan. Study of clustering validity based on fuzzy similarity. In *Proceedings of the 3rd World Congress on Intelligent Control and Automation (Cat. No. 00EX393)*, volume 4, pages 2444–2447. IEEE, 2000. (cited on Page 21)

Christopher R John, David Watson, Michael R Barnes, Costantino Pitzalis, and Myles J Lewis. Spectrum: fast density-aware spectral clustering for single and multi-omic data. *Bioinformatics*, 36(4):1159–1166, 2020. (cited on Page 27 and 30)

Sasan Karamizadeh, Shahidan M Abdullah, Azizah A Manaf, Mazdak Zamani, and Alireza Hooman. An overview of principal component analysis. *Journal of Signal and Information Processing*, 4(3B):173, 2013. (cited on Page 44)

Wisal Khan, Waqas Ahmad, Bin Luo, and Ejaz Ahmed. Sql database with physical database tuning technique and nosql graph database comparisons. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 110–116. IEEE, 2019. (cited on Page xiii and 12)

Quansheng Kuang and Lei Zhao. A practical gpu based knn algorithm. In *Proceedings. The 2009 International Symposium on Computer Science and Computational Technology (ISCSCI 2009)*, page 151. Citeseer, 2009. (cited on Page 23)

Tilman Lange, Mikio Braun, Volker Roth, and Joachim Buhmann. Stability-based model selection. *Advances in neural information processing systems*, 15, 2002. (cited on Page 21)

T Lehmann, Kay Schallert, Ramiro Vilchez-Vargas, Dirk Benndorf, Sebastian Püttker, Svenja Sydor, C Schulz, L Bechmann, A Canbay, B Heidrich, et al. Metaproteomics of fecal samples of crohn's disease and ulcerative colitis. *Journal of proteomics*, 201:93–103, 2019. (cited on Page 7, 27, 28, 32, and 33)

Ilya Levner. Feature selection and nearest centroid classification for protein mass spectrometry. *BMC Bioinformatics*, 6(1):68, Mar 2005. doi: 10.1186/1471-2105-6-68. URL https://doi.org/10.1186/1471-2105-6-68. (cited on Page 23)

Adrian S Lewis. The mathematics of eigenvalue optimization. *Mathematical Programming*, 97:155–176, 2003. (cited on Page 13)

Bin Li, J Friedman, R Olshen, and C Stone. Classification and regression trees (cart). *Biometrics*, 40(3):358–361, 1984. (cited on Page 25)

S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi: 10.1109/TIT.1982.1056489. (cited on Page 14)

J MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Math., Stat., and Prob*, page 281, 1965. (cited on Page 14)

Christopher D. Manning, Prabhakar Raghavan, and Schütze Hinrich. *Hierarchical clustering.* Cambridge University Press, 2019. (cited on Page 16)

Maria CV Nascimento and Andre CPLF De Carvalho. Spectral methods for graph clustering–a survey. *European Journal of Operational Research*, 211(2):221–231, 2011. (cited on Page xiii, 18, and 19)

Brent A Neuschwander-Tetri. Non-alcoholic fatty liver disease. *BMC medicine*, 15 (1):1–6, 2017. (cited on Page 7)

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001. (cited on Page 18 and 19)

Issa Isaac Ngom, Philippe Decloquement, Nicholas Armstrong, Raoult Didier, and Eric Chabrière. Metaproteomics of the human gut microbiota: Challenges and contributions to other omics. *Clinical Mass Spectrometry*, 14:18–30, 2019. ISSN 2376-9998. doi: https://doi.org/10.1016/j.clinms.2019.06.001. Microbiology and Infectious Disease. (cited on Page 6)

Patrick R Nicolas. *Scala for machine learning*. Packt Publishing Ltd, 2015. (cited on Page ix and 22)

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986. (cited on Page 25)

Subhash Sharma. *Applied Multivariate Techniques*. John Wiley and amp; Sons, 1996. (cited on Page 21)

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. (cited on Page 18 and 19)

Lindsay I Smith. A tutorial on principal components analysis. 2002. (cited on Page 14)

Sophia Sun. Implementing self-tuning spectral clustering, 2020. URL https://huiwenn.github.io/spectral-clustering. (cited on Page 36 and 38)

Svenja Sydor, Christian Dandyk, Johannes Schwerdt, Paul Manka, Dirk Benndorf, Theresa Lehmann, Kay Schallert, Maximilian Wolf, Udo Reichl, Ali Canbay, et al. Discovering biomarkers for non-alcoholic steatohepatitis patients with and without hepatocellular carcinoma using fecal metaproteomics. *International Journal of Molecular Sciences*, 23(16):8841, 2022. (cited on Page 27, 29, 32, 33, 34, and 46)

S. Theodoridis and K. Koutroumbas. *Pattern recognition*. Academic Press, 1999. (cited on Page 21)

Chad Vicknair, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, and Dawn Wilkins. A comparison of a graph database and a relational database. ACM Press, 2010. doi: 10.1145/1900008.1900067. (cited on Page xiii and 12)

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17 (4):395–416, Dec 2007. ISSN 1573-1375. doi: 10.1007/s11222-007-9033-z. (cited on Page ix, 9, and 10)

Lei Wang, Sujun Li, and Haixu Tang. mscrush: fast tandem mass spectral clustering using locality sensitive hashing. *Journal of proteome research*, 18(1):147–158, 2018. (cited on Page 7)

X.L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):841–847, 1991. doi: 10.1109/34.85677. (cited on Page 21)

Miin-Shen Yang and Kuo-Lung Wu. A new validity index for fuzzy clustering. In *10th IEEE International Conference on Fuzzy Systems.(Cat. No. 01CH37297)*, volume 1, pages 89–92. IEEE, 2001. (cited on Page 21)

K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001. doi: 10.1093/bioinformatics/17.4.309. (cited on Page 21)

Mohamed Zaït and Hammou Messatfa. A comparative study of clustering methods. *Future Generation Computer Systems*, 13(2-3):149–159, 1997. doi: 10.1016/s0167-739x(97)00018-6. (cited on Page 21)

Lihi Zelnik-manor and Pietro Perona. Self-tuning spectral clustering. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. URL https://proceedings.neurips.cc/paper/2004/file/40173ea48d9567f1f393b20c855bb40b-Paper.pdf. (cited on Page 18 and 19)

Pu Zhang and Qiang Shen. Fuzzy c-means based coincidental link filtering in support of inferring social networks from spatiotemporal data streams. *Soft Computing*, 22(21):7015–7025, 2018. doi: 10.1007/s00500-018-3363-y. (cited on Page 16)

Yi-Zhen Zhang and Yong-Yu Li. Inflammatory bowel disease: pathogenesis. *World journal of gastroenterology: WJG*, 20(1):91, 2014. (cited on Page 7)

I herewith assure that I wrote the present thesis independently, that the thesis has not been partially or fully submitted as graded academic work and that I have used no other means than the ones indicated. I have indicated all parts of the work in which sources are used according to their wording or to their meaning.

Magdeburg, March 28, 2023