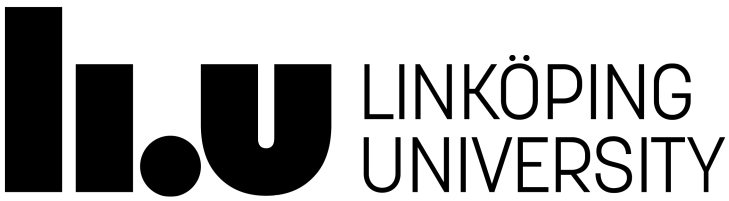
Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Statistics and Machine Learning
2023 | LIU-IDA/STAT-A–23/024–SE

Master Thesis in Statistics and Machine Learning

Multi-dimensional characterisation of multi-dataset macromolecular crystallography data

Omid Lavakhamseh



Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University
2023

Supervisors: Amanda Olmin , Nicholas Pearce
Examiner: Krzysztof Bartoszek

Contents

List of Figures

List of Tables

Abstract

The three-dimensional architecture of macromolecules, such as proteins, has become a hot topic due to its usefulness in drug discovery. However, due to limitations in microscope resolution, it is impossible to observe the constructions of protein crystals directly. Instead, crystallography is used to obtain information about the structure of protein crystals by analyzing the outputs of the experiment. Prior research has relied on atomic models to extract relevant data, yet a fraction of information is inevitably lost during the transformation from electron density to these models. Hence, this study introduces a novel approach to uncovering the three-dimensional architecture of macromolecules, employing statistical electron density analysis across multiple crystals concurrently. Initially, from a collection of 226 electron density images, the most similar images are clustered. Subsequently, the pixel values of the unit cell responsible for creating the electron density map are modeled using a Gaussian mixture model. To this end, two algorithms are introduced: one designed for two conformations and the other adaptable to a preferred number of conformations. An iterative base algorithm, utilizing both gradient descent and the Metropolis-Hastings sampling technique, is employed to analyze the images derived from crystallography experiments. The proposed model carries the potential to provide insights into the number of conformations and sub-states present in each region, as well as to identify pixels exhibiting the highest degree of structural variability.

Acknowledgments

I would like to express my deepest gratitude to my wife, Mahnaz, for her unwavering support and encouragement throughout this journey. Her constant emotional support has been invaluable to me and has helped me to overcome many challenges.

I also extend my thanks to Amanda Olmin, who generously shared her time and energy to help me whenever I encountered difficulties in my research. Her insights and guidance have been invaluable, and I could not have accomplished this without her help.

Furthermore, I would like to thank Nicholas Pearce, who generously shared his expertise and knowledge with me, particularly in areas where I needed to consult domain experts in the fields of biology and physics. His guidance has been integral to my research, and I am grateful for his support.

Finally, I would like to thank all the participants in this study who generously provided their time and resources. Without their cooperation, this research would not have been possible.

1 Introduction

Proteins are essential to many biological processes. Acquiring information about the structure of proteins can provide insight into their functions and enhance our understanding of life science and biology. Numerous researchers are utilizing machine learning models to extract valuable information about the three-dimensional structure of proteins, which can significantly contribute to drug discovery [1]. Due to the limitations of microscopy and the small size of proteins, scientists employ various experiments and analyses to infer information about protein structures.

In this study, we aim to use the output image of one of these experiments and explore machine learning methods for extracting information about the structure of proteins involved in the experiment. The main objective is to use machine learning methods to analyze the pixel values of these output images and extract meaningful information about the underlying protein structures. This analysis aims to identify the states or characteristics that generated these images. Given the complexity of the problem and the challenges involved, we need to make valid assumptions to develop the most suitable approach.

In this section, we will first review some fundamental definitions essential for understanding the concepts related to the problem and research questions. Following that, in the motivation section, we will discuss the significance and reasons behind the research. Additionally, We will outline the research questions in the aims and provide a summary of previous works conducted on this topic in the related work section.

1.1 Background

1.1.1 Protein

Proteins are composed of a sequence of amino acids. These large and complex molecules play a vital role in various biological processes that occur within living organisms, including the human body. Proteins are responsible for performing the majority of cellular functions and are essential for coordinating the activities of various tissues and organs in the body. Additionally, understanding the structure and function of proteins is necessary for developing more effective drugs [1]. Further discussion about proteins will be continued in the context of crystallization in the subsequent section, as protein crystals are a prerequisite for conducting crystallography experiments.

1.1.2 Crystallization

In the process of determining the structure of a protein, it is essential to crystallize the protein. However, there is currently no known technique for accurately predicting the specific conditions under which a particular protein will form individual crystals. The challenge in achieving appropriate conditions for protein crystallization arises from the fact that the molecules in the protein have irregular and flexible shapes, which make it challenging to organize them into a consistent, periodic crystal lattice, and there is no reliable method for predicting the specific conditions required for successful protein crystallization [2].

In the vapor diffusion method, which is the most commonly used manual technique for growing protein crystals, a small volume of the protein solution is combined with the same amount of a reservoir solution that contains a mixture of precipitation reagents. When a solution undergoes the summation of some chemical substances, called precipitation reagents, the result is the creation of a solid. After combining the protein solution with the reservoir solution, a tiny droplet of the resulting mixture is deposited onto a glass slide. The evaporation of water from the drop makes the concentrations of both the protein and precipitant gradually increase over time. Eventually, the concentration of the protein reaches a point where it exceeds its solubility limit, resulting in the formation of protein crystals. After supersaturating the protein solution, nucleation and phase separation happen, resulting in the formation of protein crystals. The state of having

a higher amount of solute in a solution than it can normally hold is referred to as supersaturation.

1.1.3 Crystal Geometry

The crystal structure consists of a recurring element in three dimensions, known as the unit cell, which serves as the fundamental building block of the crystal structure. Unit cell refers to the smallest entity within the crystal structure, which may consist of a single biological molecule, a portion of a biological molecule, or multiple biological molecules [3]. A lattice is a structural arrangement that separates space into regular units which are periodic. It is a periodic combination of the unit lattice. A unit cell is generated by the combination of unit lattice and molecular motifs, which are parts of proteins with specific structures. The periodic arrangement of finite unit cells creates a crystal. One important thing to know, which will help in this research, is that if two unit cells are identical, they are composed of the same number of molecules arranged in an identical manner [2]. Figure 1 can provide valuable insights into the nature of crystals. If we consider the entire image as a crystal, a unit cell would be a square (in this example) unit of that.

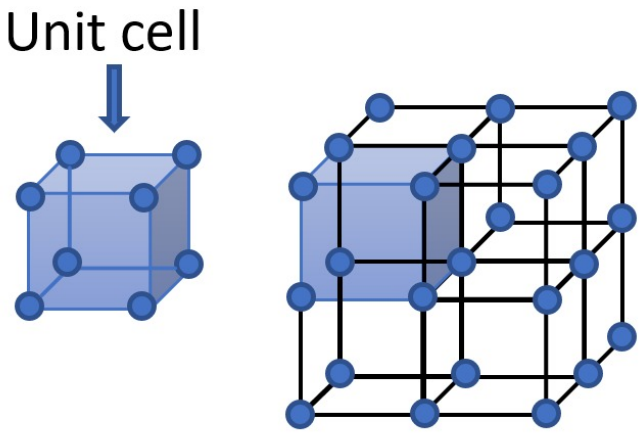


Figure 1: Crystal lattice

1.1.4 Conformation

As observed previously, it has been noted that proteins are composed of a specific sequence of amino acids, facilitating their ability to adopt unique three-dimensional conformations [4]. The term “conformation” denotes the three-dimensional structure of atoms within a crystalline structure, allowing unrestricted rotational movement without breaking any chemical bonds. Different conformations can be constructed by various factors such as temperature, pressure, chemical composition, and crystal packing. Understanding the multiple conformations present in a crystal is essential for comprehending the features and functionality of materials.

1.1.5 Crystallography

X-ray crystallography constitutes an experiment employed to ascertain the three-dimensional structural data about macromolecules like RNA and DNA. As reported by Stellato [5], X-ray crystallography has contributed to the discovery of 90% of protein structures corresponding to over 95000 structures currently stored in the Protein Data Bank. Various techniques exist for acquiring this information, but X-ray crystallography remains the most efficacious approach [5]. Alternative techniques, such as cryo-electron

microscopy (cryo-EM), have been used to determine the remaining protein structures. Although crystallography is not new, it is still used in many fields, including physics, chemistry, and biology [6]. Crystallography is a laborious and costly process [1]. In recognition of its vital role in contemporary society, the United Nations declared the year 2014 as the International Year of Crystallography. A series of coordinated global initiatives led by the International Union of Crystallography in collaboration with UNESCO aimed to increase awareness and understanding of crystallography worldwide. The slogan for the event was "Crystallography Matters." [1]

Given the limitations of conventional microscopy in visualizing molecules, scientists use crystallography as a means to acquire insights into the structural properties of molecules. Within a protein crystal, numerous identical unit cells stick together, functioning as the simulated lens necessary for investigating molecular structures [2].

Crystallographers beam X-rays toward a crystal, and then incoming X-rays get scattered by electrons in the crystal lattice. The scattered waves are combined in either a constructive or destructive manner. If they combine constructively, they reinforce each other, while destructive combinations cancel each other out. These resulting diffracted waves are then captured on a detector, typically made of phosphor or silicon, which generates a pattern. In many cases, crystals are destroyed during the experiment [7]. A diffraction pattern consists of spots of varying sizes and intensities, reflecting the arrangement of atoms in the crystal. The darker and brighter spots indicate variations in the intensity of diffracted X-rays, which provide valuable information about the distribution and density of atoms in the crystal.

The electron density map can be obtained by performing an inverse Fourier transform on the diffraction pattern. This map represents the distribution of electrons around atomic nuclei and provides valuable insights into the positions of the atoms in the crystal lattice. The resulting electron density map exhibits a cloud-like shape, representing the superimposition of different conformations of the unit cells within the crystal. In other words, it can be expressed as a weighted average of the various conformations present in the crystal's unit cells.

The result of a crystallography experiment conducted on a single molecule is not sufficiently robust. In this context, the crystal functions as an amplifier since it consists of numerous molecules and atoms. As observed in Figure 1, each crystal is composed of repeating unit cells. Each unit cell within the crystal exhibits a distinct conformation characterized by slight deviations from neighboring cells. As a result, the influence of a particular unit cell's conformation on a specific region of the electron density can be more significant than others. This phenomenon is attributed to the repeated occurrences of unit cells throughout the crystal, and it can be observed in various regions of the crystal as well.

1.2 Motivation

The motivation of this thesis is based on the paramount importance of comprehending the three-dimensional architecture of macromolecules to find intricate details about atomic positions and molecular interactions. This information is pivotal in moving forward with the creation of innovative pharmaceuticals.

Traditionally, atomic models have served as a solution for obtaining such molecular information. However, in our study, we attempt to depart from this conventional approach, choosing instead to analyze the electron density of these macromolecules. The transformation of electron density data into atomic models could potentially lead to a loss of vital information. By conducting a direct and comprehensive electron density analysis, we aim to preserve all the intrinsic details that might be diluted in the transformation process. By analyzing these electron density pixel values, we seek fresh insights and perhaps discover uncharted facets of molecular structure that might be obscured within atomic models.

[1] https://zenodo.org/record/48770#.ZG1SQnZBy5c

Moreover, the scope of our investigation extends beyond the analysis of a single crystal structure. We propose to analyze multiple crystal structures together, which is the reason for the term multi-data set in the subject of the thesis. Examining multiple similar crystal structures could enhance the accuracy and reliability of our findings.

In conclusion, this novel approach to studying electron density pixel values in multiple crystals could offer a more precise and detailed understanding of the structure of macromolecules, which can catalyze the development of new pharmaceutical solutions.

1.3 Aim

This research endeavor aims to utilize diffraction pattern images to extract detailed information about macromolecules' three-dimensional architecture.

Firstly, we aim to derive meaningful electron density maps from the given diffraction patterns. By translating these patterns into electron density maps, we desire a comprehensive image of the macromolecular structure, providing a clearer understanding of the molecular complexities.

Secondly, our objective extends to analyzing multiple crystal electron density maps. We intend to solve information concerning their unit cell conformations. This comparative study could excavate relationships or differences among the multiple crystals, enriching our understanding of the macromolecular structures.

Finally, we aim to leverage these electron density maps and analyses to generate a precise three-dimensional structure of protein crystals. We plan to tackle this by formulating and addressing key research questions that can direct our understanding toward a more defined and accurate structural model.

1. Can multi-dataset analysis reveal how many sub-states are present in each region?

As previously stated, electron density can be perceived as the product of the superimposition of conformations within unit cells. These unit cells may maintain identical or dissimilar conformations. In order to clarify this aspect, it is essential to investigate the quantity and variety of sub-states, or conformations, contributing to the formation of each pixel within the electron density map.

2. Can multi-dataset analysis reveal which sub-states are present in each region?

To clarify, the second research question asks whether multi-dataset analysis can provide information on which sub-states (or conformations) are present in each region of the electron density map. In other words, can we use statistical analysis to identify the different conformations contributing to a particular electron density map region?

3. Which map regions are most variable, and how does this correlate to structural variability?

In this research question, we aim to analyze the pixel values of the electron density map to determine which regions are the most variable. The term "region" refers to a section of the electron density image that a group of pixels can form. By examining the relationship between the neighboring pixels, we can identify the degree of variability in each region and understand how it correlates with the structural variability of the macromolecule.

4. Can long-distance correlations be detected between different regions?

After obtaining information about the characteristics of each pixel in the electron density map, the next question is detecting long-distance correlations between different regions. This involves analyzing the relationships between non-adjacent pixels and determining if they are correlated.

1.4 Literature Review

Nowadays, there is a growing demand for the application of machine learning in crystallography. While several studies have employed statistical analysis and machine learning to derive insights into protein crystal structures, to the best of our knowledge, our approach is novel in its simultaneous analysis of multiple electron density maps to extract such information. Previous works in this field have primarily focused on other techniques or alternative methods to extract information about protein structure. The following sections will discuss previous research studies and their utilization of machine learning to analyze the outcomes of crystallography experiments.

In the study by Souza [6], three methodologies – Random Forests, Support Vector Machines, and a CNN based on ResNet50 – were employed to classify diffraction images. As previously mentioned, specific diffraction images are unsuitable for further investigation. One of the classification methods used in this study aimed to differentiate images containing diffraction patterns from those lacking such patterns. This approach enables crystallographers to avoid wasting time on low-quality data.

In another study by Venkatraman [8], a combination of machine learning and deep learning models was employed to predict space groups. Space groups refer to the theoretical framework that describes the arrangement of atoms within a three-dimensional crystal lattice.

The authors of [1], stated that the process of crystallography and NMR could be both laborious and costly. To overcome these challenges, they utilized an exponential method for obtaining information about protein structure. Additionally, they noted that many researchers are currently exploring suitable models for determining protein functions using this approach.

In the research conducted by Shapovalov and Dunbrack [9], statistical analysis is applied to analyze the electron density of a large set of proteins. The analysis focuses on exploring the conformational properties of protein side chains, which play a crucial role in establishing specific interactions between proteins and other molecules. The study also compares two distinct side chains to gain further insights into their structural characteristics.

2 Data

This section is dedicated to data processing and visualization. It involves a detailed analysis of the data and the execution of essential procedures to ensure its suitability for analysis. These procedures include filtering, performing the inverse Fourier transform, aligning images, and flattening the data. In addition, the section concludes with a data visualization step that offers a comprehensive overview of the dataset.

2.1 Data Source

The dataset [2] comprises 226 diffraction pattern images stored as MTZ files. These files were acquired by conducting crystallography experiments on 226 distinct crystals. Each includes atomic model (PDB) and data file (MTZ).

2.2 Data Description

The present study encompasses 226 MTZ (merged reflections) files generated through X-ray crystallography experiments conducted on 226 distinct protein crystals. As previously noted, the diffraction pattern captures information pertaining to the position, intensity, and phase of diffracted waves. This information is subsequently utilized to derive the electron density map, which provides insights into the positions of atoms within the crystal lattice. To analyze the MTZ files, the Coot software [10] is utilized, and Figure 2 illustrates a sample output. It should be noted that Coot performs Fourier transforms on the diffraction pattern, yielding a portion of the electron density map. Indeed, the image can be perceived as a circular segment of the electron density map that is visible through a camera lens with a circular shape.

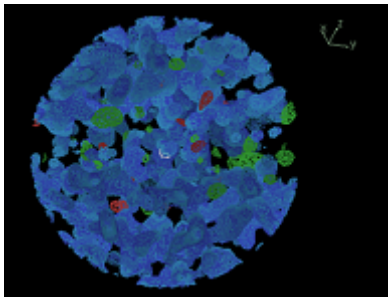


Figure 2: A circular segment of the electron density map

In this thesis, the CCP4 [3]package was employed, specifically utilizing the Gemmi library to read MTZ files using Python. The resulting data was stored in a dataframe (Figure 3). The dataframe contains information about each reflection, with each row detailing the Miller indices (h-k-l), which are indexes of reciprocal space, intensity (2FOFCWT), and phase (PH2FOFCWT).

2.3 Data Preprocessing

2.3.1 Filtering

This study aims to determine the positions of atoms by analyzing corresponding values in different MTZ measurements. The h-k-l values represent three dimensions that indicate the position of each reflection in diffraction space (also known as reciprocal space). Initially, the diffraction points present in all patterns (measurements) need to be filtered. This step entails comparing the availability of points (based on their h-k-l positions) across all images and removing any points not present in all. Notably, these points may have different phases and intensities. As a result, 13446 spots are retained, representing the diffraction points that exist in all images.

[2]https://zenodo.org/record/48770#.ZG1SQnZBy5c
[3]Source: https://www.ccp4.ac.uk/html/mtzformat.html

	H	K	L	FreeR_flag	F	SIGF	FC	PHIC	FC_ALL	PHIC_ALL	2FOFCWT	PH2FOFCWT
0	0.0	0.0	4.0	2.0	NaN	NaN	20890.607422	180.000015	3323.593750	180.000061	3323.593750	180.000061
1	0.0	0.0	8.0	13.0	2270.359863	27.887918	573.453125	359.999969	1146.901611	360.000000	3393.818115	360.000000
2	0.0	0.0	12.0	18.0	3427.665527	42.041695	4873.706543	179.999985	3219.302734	179.999969	3636.028320	179.999969
3	0.0	0.0	16.0	6.0	NaN	NaN	3865.468506	180.000015	3165.577148	180.000015	3165.577148	180.000015
4	0.0	0.0	20.0	6.0	1080.997314	19.234512	221.217300	0.000000	63.633255	0.000362	820.633972	0.000362
...	...	...	...	...	...	...	...	...	...	...	...	...
13441	28.0	0.0	1.0	16.0	262.330780	5.346642	258.792816	315.000000	247.447525	315.000000	277.214050	315.000000
13442	28.0	0.0	2.0	8.0	469.801300	4.887661	516.117371	90.000000	501.171997	90.000000	438.430603	90.000000
13443	28.0	1.0	0.0	19.0	446.478485	4.875942	388.013275	0.000009	381.621490	0.000010	511.335480	0.000010
13444	28.0	1.0	1.0	2.0	79.483101	9.284325	19.389311	6.132946	21.212502	7.140066	36.835293	7.140066

Figure 3: Diffraction pattern data derived from MTZ files

2.3.2 Fourier Transform

As observed in Figure 2, the Coot application is utilized to open the MTZ file, which in turn displays the electron density. This process involves performing an inverse Fourier transform of the diffraction pattern, which results in the electron density distribution visualization. The information displayed in Coot pertains to the electron density of the crystal structure, specifically the electron density map derived from the diffraction pattern. In order to obtain a 3D representation of the electron density in real space, we aim to perform a transformation that allows us to visualize the electron density of the corresponding diffraction pattern. This section begins by providing some fundamental definitions and then explains the process of generating an electron density map.

The Fourier transform is a technique that enables the analysis of a waveform, such as a function or signal, by decomposing it into a different representation consisting of various sine and cosine functions with distinct frequencies. In other words, it is a mathematical operation that transforms a function into a representation that characterizes the frequencies contained within the function. The Fourier transform serves as both a mathematical operation and a means of representing the original function in the frequency domain [11].

Consider that the signal in the time domain is represented by f(x), whereas the signal in the frequency domain is represented by F(S). The Fourier transform of f(x) can be represented by a combination of real and complex numbers, as referred to by F(s). Fourier transform for a continuous function is as Equation (1).

$$F(s) = \int_{-\infty}^{+\infty} f(x) e^{-2\pi i s x} dx \tag{1}$$

The Fourier transform is an invertible function, meaning that f(x) can be obtained from F(s) by applying the inverse Fourier transform. Inverse Fourier transform for a continuous function can be obtained as Equation (2).

$$f(x) = \int_{-\infty}^{+\infty} F(s) e^{2\pi i s x} ds \tag{2}$$

Real space is the space in which we live in the physical universe. It encompasses the three dimensions of length, width, and height we perceive daily. It is the fundamental space where we exist and interact with our surroundings. The concept of real space is essential for comprehending the physical world and the various interactions that occur within it. In real space, we generally define things in terms of positions and distances.

The concept of diffraction, rooted in constructive and destructive interference, is aptly portrayed within reciprocal space. The idea of reciprocal space was presented by J. W. Gibbs in 1881. It fundamentally relies on the notion that planes within real space correspond to points within reciprocal space [12]. As the term indicates, it incorporates

the concept of reciprocals relating to real space distances. Reciprocal space is an imaginary space where atom planes are symbolized by reciprocal points, which is the transformation of real space [13]. Reciprocal space serves a role comparable to the frequency domain that emerges from applying the Fourier transform to a function of time.

In crystallography, each crystal structure is associated with two lattices: the crystalline lattice in real space and the reciprocal lattice in Fourier space. The reciprocal lattice can be visualized as the diffraction pattern obtained from the crystal, while the crystalline structure is represented by a microscopic image in real space. A crystal lattice represents the positions of the atoms and the distances between them. Reciprocal space describes the periodicity or repeating nature of the crystal structure. It is important to note that the lattice in Fourier space, which is the reciprocal lattice, corresponds to the reciprocal of the crystalline lattice in real space [14]. In the other words, The Fourier transform facilitates the transition from reciprocal space to real space, with h, k, and l representing the dimensions in reciprocal space and x, y, and z representing the dimensions in real space.

To understand how the inverse Fourier transforms diffraction patterns into electron density maps, Figure 4 (left image) can be helpful. This graph is a shape of three delta functions in which the x-axis represents frequency, and the y-axis represents magnitude in the frequency domain.

Upon applying the inverse Fourier transform to these three delta functions, we obtain a composite sine wave, as illustrated in Figure 4 (right image). This wave constitutes the summation of three individual sine waves, each characterized by distinct frequencies and amplitudes. In this depiction, the x-axis represents time, and the y-axis denotes amplitude [15].

Now consider the delta graphs in Figure 4 as the diffraction pattern, which has intensity and frequency. Figure 4 (left image) represents frequencies in diffraction patterns based on the intensities that have the most impact on building the electron density. By taking IFT of them in 3D, we will have a continuous 3D cloudy shape like what we have seen before as electron density in Figure 2.

To gain insights into the roles of amplitude and phase in the Fourier transform formula, Equation (2) can be rewritten in terms of the variable w, resulting in Equation (3). In this equation, $F(w)$ is defined by Equation (4), where $\phi(w)$ represents the phase and $A(w)$ represents the amplitude.

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(w) e^{iwx} dw \tag{3}$$

$$F(w) = A(w) e^{i\phi w} \tag{4}$$

So, $f(x)$ can be expressed as:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} A(w) e^{i(wx + w\phi)} dw \tag{5}$$

Equation (5) illustrates the Fourier transform of the function $f(x)$ utilizing the amplitude $A(w)$ and the phase ϕ. It highlights the connection between the real space and the frequency domain (reciprocal space), where the integral of $A(w)e^{i(wx+w\phi)}$ captures the contributions of different frequencies, including their respective amplitudes, in constructing the function $f(x)$.

$$\rho(x, y, z) = \frac{1}{V} \sum_{hkl=-\infty}^{\infty} |F(hkl)| \cdot e^{-2\pi i[hx+ky+lz-\phi(hkl)]} \tag{6}$$

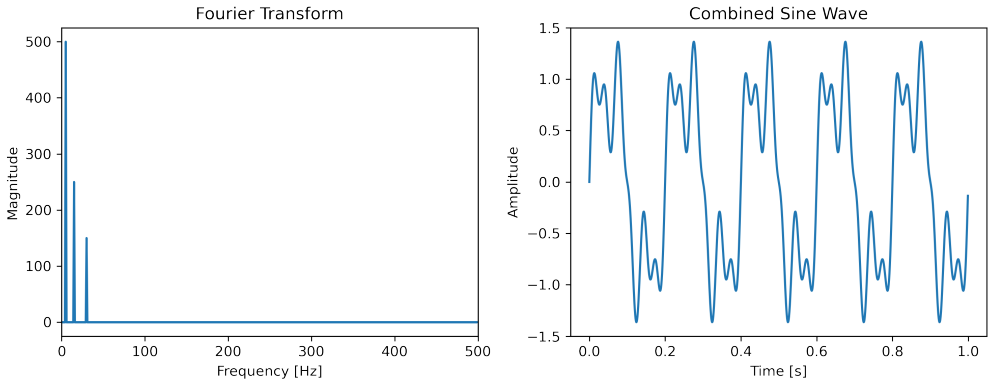


Figure 4: Diffraction pattern(left) electron density map(right)

Equation (6) illustrates the Fourier transform relationship between real and reciprocal space. It involves the intensity of diffraction spots (F(hkl)), the Miller indices (h, k, l) representing spot locations, the phase of the captured wave ($\phi(hkl)$), and the volume of the unit cell (V). This highlights the holistic nature of diffraction, where calculating the electron density at a specific point in space (xyz) necessitates considering the contributions from all diffraction spots generated by the crystal.

2.3.3 Alignment

In order to compensate for the slight misalignment of electron densities, an alignment procedure is necessary. It should be noted that the images are only slightly rotated and misaligned in comparison with each other, and they are all in the same position. The SimpleITK library's sitk function is utilized for the purpose of alignment, where one of the images is taken as the reference or fixed image, and the remaining images are rotated and aligned to match it.

2.3.4 Flattening

After completing the above steps, the result is an array with a shape of $226 \times 60 \times 60 \times 120$. 226 is the number of measurements, and three other dimensions represent the number of pixels in each image. To facilitate the analysis of the data, it is advisable to flatten the array and convert it into a dataframe. This results in a dataframe with 226 rows and 432000 columns, where each row represents information about a specific measurement, and each column contains the corresponding value of a pixel.

The dataframe in Figure 5 has been achieved after all the steps above. It contains 226 rows and 432000 columns. The rows represent the electron density images, while the columns represent the corresponding electron density pixel values.

	Pixel_1	Pixel_2	Pixel_3	Pixel_4	Pixel_5	Pixel_6	Pixel_7	Pixel_8	Pixel_9	Pixel_10	...	Pixel_431991	Pixel_431992	Pixel_431993
D1	0.300000	0.040000	-0.270000	-0.280000	-0.470000	-0.310000	-0.060000	-0.160000	-0.390000	0.250000	...	0.490000	0.320000	-0.320000
D2	-0.029921	0.019818	-0.200119	-0.350066	-0.479891	-0.439690	-0.189920	-0.200050	-0.319442	0.270484	...	0.369711	0.049611	-0.379584
D3	-0.019596	-0.109822	-0.439545	-0.529029	-0.709169	-0.449427	-0.159241	-0.209545	-0.399831	0.239709	...	0.359973	0.169925	-0.349902
D4	0.089988	0.009961	-0.250009	-0.310015	-0.409994	-0.369962	-0.119994	-0.080029	-0.269921	0.250090	...	0.479635	0.229646	-0.309917
D5	0.229859	0.029788	-0.270050	-0.340120	-0.509908	-0.379816	-0.120035	-0.170163	-0.399604	0.160410	...	0.559740	0.339370	-0.389725
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
D222	0.290177	-0.079808	-0.469399	-0.498909	-0.629149	-0.369544	-0.119320	-0.129564	-0.449788	0.109815	...	0.509995	0.369975	-0.309981
D223	0.170248	-0.129707	-0.599004	-0.418477	-0.538824	-0.479156	-0.128885	-0.159285	-0.349686	0.249644	...	0.389980	0.249905	-0.399909
D224	0.300201	-0.169689	-0.569374	-0.439128	-0.639277	-0.449569	-0.139385	-0.129647	-0.379935	0.149738	...	0.449951	0.310007	-0.299968
D225	0.190279	-0.169648	-0.509326	-0.409012	-0.599192	-0.429542	-0.109367	-0.119622	-0.369920	0.179664	...	0.459940	0.340008	-0.279959
D226	0.150204	-0.219705	-0.589315	-0.459023	-0.629191	-0.439523	-0.089374	-0.129617	-0.399842	0.099835	...	0.439988	0.289954	-0.309956

Figure 5: Flattened data

Since a proper understanding of the data requires some guidance, Figure 6 has been

used to illustrate it, as we have already become familiar with the concepts of protein and unit cells in the introduction section. Assume that the protein crystal is represented in Figure 6a, and the squares within it represent the unit cells. Figure 6b represents the corresponding electron density for that protein crystal. The value of pixel_1 in the electron density map (Figure 6b) is a weighted average of the corresponding pixels in all of the unit cells.

We will now return to the dataset, which consists of a dataframe with a shape of 226×432000. As previously mentioned, each row contains information about the pixels in an electron density image. Therefore, the dataframe contains information about the values of pixels in electron density. In the following sections, we will demonstrate how we can use these pixel values to obtain information about the protein crystal depicted in Figure 6a. It should be noted that this simulation only involves 16 unit cells, whereas, in reality, there are trillions of unit cells in a crystal lattice. In this case, we are dealing with 432000 pixels, which means that there are 432000-pixel values in the right image (although only 12 are shown in the plot).

Figure 7 shows the histogram of the first six pixels, indicating that the distribution of these pixels closely follows a normal distribution. Figure 8 shows the correlation coefficients of 10 randomly selected pixels between 10 and 20. The information from Figure 5 was utilized to obtain these coefficients, specifically focusing on the columns corresponding to pixels 10 to 20 across all the measurements. The results show a gradual change in the correlation values, with the nearest pixels demonstrating higher correlation levels than the rest, as expected. Similar patterns would be observed when examining the correlation of other neighboring pixels.

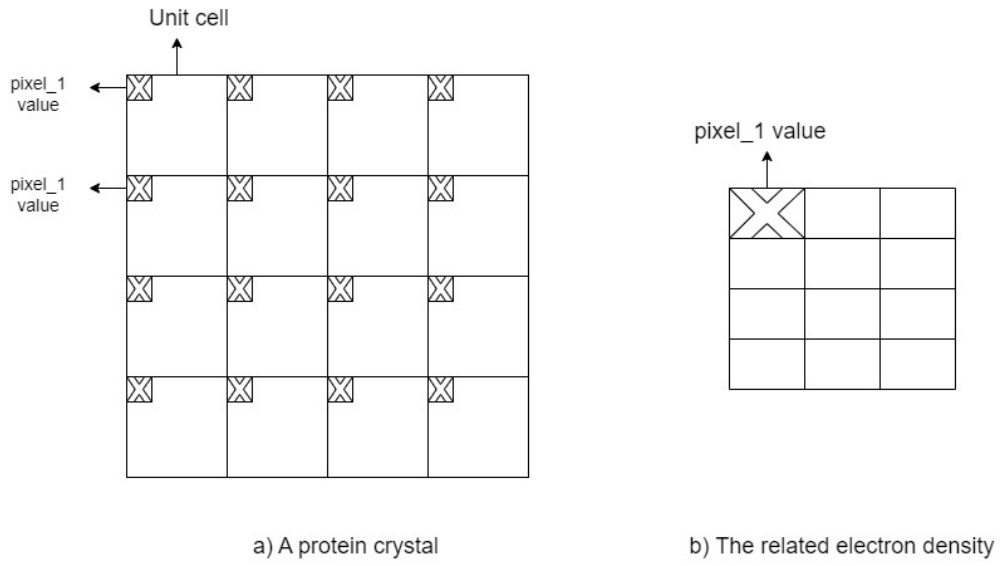


Figure 6: Schematic representation of protein and electron density. In image (a), unit cells of the protein crystal are depicted, with highlighted pixels representing the original values that contribute to the creation of the first pixel in the corresponding electron density map is shown in image (b).

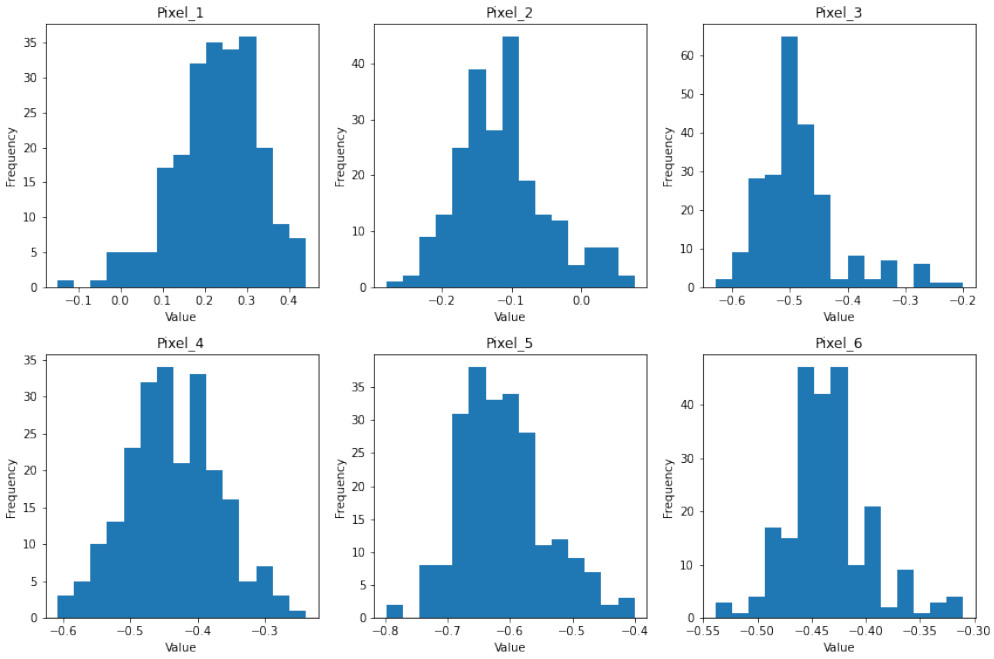


Figure 7: Histogram of the first six pixels demonstrating a normal distribution pattern

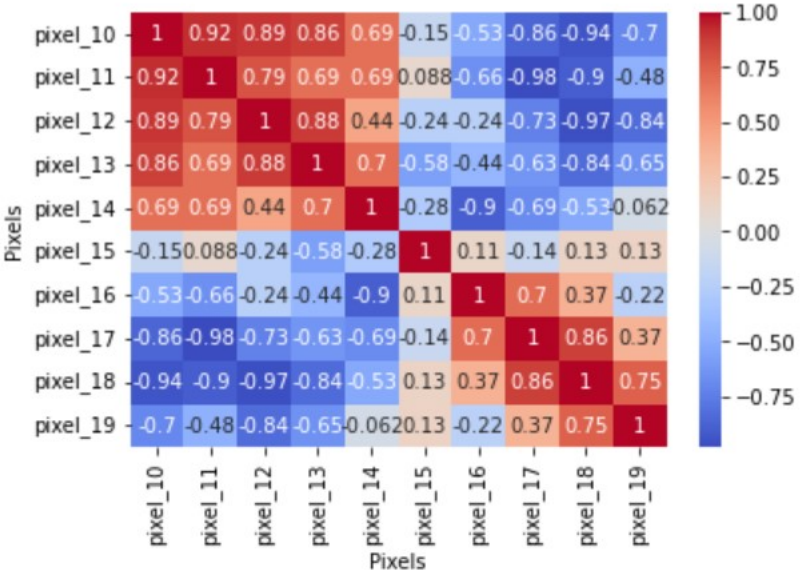


Figure 8: Correlation coefficients heatmap

3 Theory

In this section, we provide a comprehensive explanation of the theoretical concepts and methodologies that form the basis for the subsequent sections of this thesis. We begin by presenting an overview of feature extraction methods, particularly emphasizing Principal Component Analysis (PCA). Next, we delve into two clustering techniques and explore several distributions that play a crucial role in the proposed methodology. We then introduce the Gaussian mixture model and the Mean Squared Error (MSE) function, which serves as a metric to assess the accuracy of the proposed approach. Lastly, we discuss the Metropolis-Hastings algorithm and its significance within the context of this research. These theoretical foundations serve as the underpinning for the following sections, where we apply these concepts to analyze and extract valuable insights from our dataset.

3.1 Feature Extraction

In various fields, especially in the domains of biology and chemistry, there has been a significant increase in the availability of data. However, this abundance of data poses certain challenges. One of the primary challenges is the presence of numerous features associated with specific experiments. Working with such large datasets not only presents difficulties in terms of data manipulation and processing but also imposes significant computational and time constraints. Moreover, a substantial portion of these features may not contribute significantly to the analysis, making their inclusion unnecessary. Moreover, the high dimensionality of the data increases the risk of overfitting the training data, making it unfavorable to work with all original features. Hence, it is not advisable to utilize the entire set of original data [16].

There are two approaches for reducing the number of features: feature selection and dimensionality reduction.

The first way of feature extraction is feature selection. It is about selecting some features that are more important than others and assuming that these features can represent most of the information in the data [17]. There are several methods for feature selection, including:

1. Ignoring the feature with constant values or low variation, assuming they do not give us any information about the data.
2. The features that have many missing values are ignored, as it is not recommended to perform missing value imputation for this case. Therefore, we need to remove those features.
3. Do not need to consider features highly correlated with each other as one can explain the information of the others.

Dimensionality reduction is another technique that converts the feature dimension to a lower dimensional space [17]. One of the most beneficial dimensionality reduction methods is PCA.

3.1.1 PCA

The methodology of principal component analysis was introduced by Karl Pearson [18] and later developed by Hotelling [19]. PCA is a widely utilized technique employed to tackle the complexities associated with high-dimensional datasets. Its primary objective is to reduce the number of features through the creation of new features called components. Therefore enhancing interpretability while minimizing information loss. By generating new uncorrelated variables known as principal components, PCA effectively captures and maximizes the variance present in the data [20].

Indeed we are mapping data or representing them in other features. The first principal component explains the most variation in the data points, and subsequent components

explain the next highest variations. These new features are independent. Therefore, this method helps remove variables that are correlated with each other.

3.2 Clustering

Using clustering methods, we can group similar data to the same group so data points in the same cluster have the most similarity. While it is possible to visually cluster 1D, 2D, and 3D data points, clustering by visual inspection becomes more difficult as the dimensionality of the data increases.

Based on how data points are assigned to clusters, clustering can be performed using two primary methods: hard and soft. In hard clustering, as the name suggests, data points are assigned exclusively to a single cluster. In contrast, soft clustering assigns a probability or likelihood to each data point, indicating its potential membership to each component or cluster.

All clustering algorithms search for similarities and differences in data points to group them. There are four different methods of performing clustering: 1) Partitional, such as k-means; 2) Hierarchical; 3) Density-based, such as DBSCAN; and 4) Distribution-based.

3.2.1 K-means

The k-means clustering was proposed by Lloyd [21]. The k-means algorithm is known for its straightforward implementation and easy comprehension. Its fundamental principle revolves around minimizing the variation within each cluster by defining clusters appropriately.

The k-means clustering algorithm takes a set of data points and the number of clusters, which is typically denoted as "k" [22], but for consistency with the notation used in the methods section, it is denoted as "M" here.

The k-means clustering algorithm can be summarized as follows:

- Step 1: Selecting the number of clusters, denoted as M.
- Step 2: Choosing M initial points, each representing a cluster.
- Step 3: Measuring the Euclidean distance between each data point and the initial centroids selected in the previous step. Assign each data point to the closest cluster.
- Step 4: Calculate the mean of the data points within each cluster, known as the centroid.
- Step 5: Calculate the Within-Cluster Sum of Squares (WCSS) using the provided formula, which measures the total variation within each cluster. In Equation (7), the variable M represents the number of clusters, the variable k denotes the cluster number assigned to each data point, ranging from 1 to M, and the variable n signifies the total number of data points within each cluster.

$$WCSS = \sum_{k=1}^{M} (\sum_{d_i \ in \ C_k}^{n} (\ distance(d_i, C_k)^2)) \tag{7}$$

- Step 6: Iteratively perform steps 3 to 5 until convergence is achieved, which occurs when the centroids no longer changes.

3.2.2 DBSCAN

DBSCAN is a density-based clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu [23]. DBSCAN is the abbreviation for Density-based spatial clustering of applications with noise, which clusters data points by defining dense regions. It is widely recognized as one of the most popular density-based clustering algorithms [24].

It has two parameters:

- ϵ represents the distance metric that defines the neighborhood. It corresponds to the radius of a circle, and all data points within this circle are considered neighbors of a given point, denoted as p.
- minPts refers to the minimum number of samples required for a point to be classified as a core point. In other words, if a point has at least minPts neighbors within its ϵ-radius, it is considered a core point.

DBSCAN operates according to the following steps:

- Step 1: It initiates by selecting a data point and finding all the neighbor points within ϵ.
- Step 2: The algorithm checks minPts value and determines if the current point is a core point. If so, it forms a cluster.
- Step 3: Then proceeds to the subsequent unvisited points. The process is repeated until all points are assigned to a cluster.

3.3 Dirichlet Distribution

The Dirichlet distribution [25] is a continuous multivariate probability distribution. It is a generalization of the beta distribution in multivariate form. It is frequently indicated as Dir(α) with concentration parameters α. Each element in the α is denoted as α_i. In Bayesian statistics, the Dirichlet distribution is regularly employed as the conjugate prior for both categorical and multinomial distributions. The Dirichlet probability distribution function is as Equation (8), in which the variable x_i represents a set of K variables.

$$f(x_1, ..., x_k; \alpha_1, ..., \alpha_k) = \frac{1}{\beta(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \tag{8}$$

$$\beta(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\alpha_0)} \tag{9}$$

$$\alpha_0 = \sum_{i=1}^{K} \alpha_i \tag{10}$$

K is the number of variables and must be larger than one, and α is a vector parameter of positive values.

The vector x, which consists of positive values x_1 to x_K, represents a simplex, characterized by the constraint:

$$\sum_{i=1}^{K} x_i = 1 \tag{11}$$

Dirichlet distribution is utilized for modeling categorical data in which results belong to various distinct categories. Using the simplex property enables us to use this distribution to model numbers that are percentages and proportions.

3.4 Gaussian Distribution

The normal, or Gaussian distribution is the most commonly utilized continuous probability distribution [26]. The Gaussian distribution is a bell curve distribution where the majority of values are centered around the distribution mean. The probability density function (PDF) for a Gaussian distribution is as Equation (13):

$$X \sim N(\mu, \sigma^2) \tag{12}$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{13}$$

Where μ and σ are the mean and standard deviation (the dispersion of data from the central tendency) of the distribution, respectively.

The normal distribution is important in various contexts. It is useful in situations where we lack information about the underlying distribution of a natural event. For instance, in the field of medical science, many events or experiments can be approximated by a normal distribution. One such example is the height of individuals within a specific gender population, such as men or women. In a normal distribution, the distribution curve is symmetrical, and the mean, median, and mode are all equal [27]. Another important feature is that the form of distribution remains unchanged even if we change the mean and variance, so we can always visualize a normal distribution as a bell-shaped curve around its mean. Based on the Empirical rule, nearly all values (99.7%) of observations are expected to fall within the first three standard deviations of the mean($\mu \pm 3\sigma$).

3.5 Central Limit Theorem

According to the Central Limit Theorem(CLT) [26], if we have n random samples, denoted as X_1, X_2, ..., X_n, from an unknown distribution with a mean of μ and a variance of σ^2, $\bar{X}$ is a sample mean as defined in Equation (14).

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{14}$$

As the sample size increases, the distribution of the sample mean $\bar{X}$ approaches a normal distribution with a mean of μ and a variance of σ^2/n. The central limit theorem allows us to make conclusions about the population based on sample data, even without knowledge of the population distribution. This important result is known as the central limit theorem. The sufficiency of the sample size in approximating a normal distribution is influenced by the similarity between the population distribution and the normal distribution [27].

3.6 Gaussian Mixture of Distributions

As demonstrated in the previous example, the Gaussian distribution is effective in modeling the height of women and men individually. However, if the objective is to determine a distribution that encompasses both genders, a challenge arises. Upon plotting the data points, it becomes evident that the data exhibits two prominent clusters with distinct means. Therefore, the limitations of a Gaussian distribution prevent it from adequately representing the entirety of the data points. In such cases, as suggested by Bishop [28], a possible approach is to utilize a linear combination of two Gaussian distributions. The resulting distribution, which is a probabilistic model, is referred to as a mixture of distributions. In this specific example, we utilized a mixture of Gaussian distributions. For the example with height, we observed two distinct groups, leading to a combination of two Gaussian distributions. However, in different scenarios, a sufficient number of models or a linear combination of numerous distributions may be necessary. In fact, by modifying the mean, covariance, and mixing coefficients, we can closely approximate any continuous distribution [28]. The formulation of a mixture of M Gaussian distributions is expressed as follows:

$$p(x) = \sum_{k=1}^{M} \pi_k N(x|\mu_k, \Sigma_k) \tag{15}$$

where $N(x|\mu_k, \Sigma_k)$ represents the Gaussian distribution with mean μ_k and covariance matrix Σ_k, and π_k denotes the corresponding mixing coefficient. The mixing coefficients need to satisfy the following conditions:

$$0 \leq \pi_k \leq 1 \tag{16}$$

$$\sum_{k=1}^{M} \pi_k = 1 \tag{17}$$

3.7 Least Squares Errors

The information provided in the following discussion is sourced from the work of Wackerly [26]. The objective of this study is to predict the actual values (Y) for a particular phenomenon. These predicted values are denoted as $\hat{Y}$. The methodology employed in this study involves the fitting of a line to a given dataset, where the error term ϵ is derived from a probability distribution with an expected value of zero ($E(\epsilon) = 0$).

$$Y = \beta_0 + \beta_1 x + \epsilon \tag{18}$$

The line predicts values of Y:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{19}$$

The goal is to minimize the discrepancy between the data points and the line that is fitted to them. $\hat{y_i}$ is the predicted value of ith point and when $x = x_i$ (Equation (20)). Error is the difference between y_i and $\hat{y_i}$(Equation (21)).

$$\hat{y_i} = \hat{\beta}_0 + \hat{\beta}_1 x_i \tag{20}$$

$$e = y_i - \hat{y_i} \tag{21}$$

By considering all data points, the objective is to minimize the sum of squares of the errors, as shown in Equation (22):

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y_i})^2 = \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2 \tag{22}$$

Similarly, the MSE can be introduced, which represents the average of the squared errors, as expressed in Equation (23):

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y_i})^2}{n} \tag{23}$$

3.8 Metropolis-Hastings Algorithm

One of the commonly used Markov Chain Monte Carlo (MCMC) algorithms is the Metropolis-Hasting algorithm. The Metropolis algorithm is proposed by Nicholas Metropolis [29]. It is a technique for generating a series of random samples from those distributions that you can not sample from directly. We use this technique to indirectly sample from them and use it to represent the probability distribution. So Metropolis-Hasting algorithm, like other MCMC algorithms, produces a sequence of random samples x_t, t=1, . . . , T.

In the Metropolis-Hasting algorithm, we have two main functions, the function $f(x)$ is called target distribution and represents the probability distribution from which we intend to draw samples from [30]. $q(x)$ is called proposal distribution, and it is chosen based on a distribution that should be the most similar to the target distribution. The entire process of the algorithm can be observed in the following three steps.

- Step 1:

It is an iterative algorithm that begins with an initial value of x_1, chosen to be within the support of the target distribution. The other values of x_t are generated iteratively by sampling from proposal distribution(y_t) based on the acceptance ratio, which will be discussed in step 3. Generating a sample y_t, which is called proposal from the proposal distribution with a density function of $q(y_t|x_{t-1})$, where x_{t-1} is the previously obtained value in the sequence.

- Step 2:

Calculating the acceptance probability, denoted as $p(t)$, can be done using the following formula:

$$p(t) = \min(\frac{f(y_t)}{f(x_{t-1})}\frac{q(x_{t-1}|y_t)}{q(y_t|x_{t-1})}, 1) \tag{24}$$

It is important to mention that if the proposal distribution follows a normal distribution, the acceptance probability, denoted as $p(t)$, will be adjusted due to the symmetry of the normal distribution. The modified acceptance probability is calculated as follows:

$$p(t) = \min(\frac{f(y_t)}{f(x_{t-1})}, 1) \tag{25}$$

In this context, the choice of the variance of the normal distribution, often referred to as the step size, can play a significant role.

- Step 3: Checking the acceptance-rejection criteria

In this stage, a random value, denoted as u_t, is sampled from a uniform distribution. The acceptance-rejection criterion is then evaluated as follows: If $u_t \leq p_t$, the proposal is accepted, and the current value of x_t is updated with the value of y_t. Otherwise, the proposal is rejected if the above criterion is not satisfied, indicating that $u_t > p_t$. In this case, the current value of x_t remains the same as the previous value x_{t-1}, as shown by the following equation: $u_t > p_t \rightarrow x_t = x_{t-1}$

We will get higher acceptance probabilities if the proposal is more similar to the target distribution. x_t and x_{t+1} are typically not independent. However, their dependence weakens as the number of iterations increases, making them increasingly closer to independence. The distribution of the sequence of x_t gets closer to the target distribution.

4 Methodology

After providing a theoretical foundation, we will now delve into the methodology. In this section, the first step is applying clustering algorithms to the measurements to identify the most similar data points for further statistical analysis. Subsequently, we will present several approaches, each of which necessitates specific assumptions to address the research problem effectively.

4.1 Clustering

We have already introduced some clustering methods in the theory section. The objective of this step is to eliminate measurements(images) dissimilar to most of the measurements, and we have selected two clusters for this purpose. The performance of different clustering models will be evaluated to determine which achieves this goal most effectively. As the study focuses not on clustering, a few types of clustering are selected, and their results are compared to identify the most suitable method for further analysis. For this purpose, three clustering methods, including k-means, DBSCAN, and GMM, are utilized. One hundred seventy-one images have been grouped through the clustering process. More detailed results will be presented in the results section.

4.2 Data Partitioning

Following the clustering of the measurements, it is necessary to split the data into training and testing sets to evaluate the models. To achieve this, the data was first reshuffled to prevent dependence on the position of data points. Then, 70% of the available data was randomly assigned to the training set, while the remaining 30% was assigned to the testing set.

4.3 Statistical Analysis

This subsection begins with a concise example that serves to illustrate the underlying process. Subsequently, a systematic presentation of more generalized solutions follows. The utilization of an initial example is essential in simplifying the understanding of the main algorithm, particularly due to the inherent complexity of the problem at hand.

Suppose a protein crystal has two distinct conformations labeled A and B, as illustrated in Figure 9. In each unit cell, only one of these conformations is present. We have 16 unit cells in this example, but a protein crystal can contain trillions of unit cells. In this case, we have 8 unit cells with conformation A and 8 unit cells with conformation B. Each of these conformations has a unique value in their first pixels. In Figure 9b, P_1 represents the first pixel of the electron density map, and it is known that each pixel in the electron density map corresponds to the weighted average of the relevant pixels in the protein crystal.

$$P_1 = W_A A_1 + W_B B_1 \tag{26}$$

W_A is the proportion of unit cells with conformations A, and W_B is the proportion of unit cells with conformation B. A_1 is the value of the first pixel in the conformation A, and B_1 is the value of the first pixel in conformation B.

One could argue that P_1 is always the weighted average of two numbers (in this example for two conformations), namely A_1 and B_1, and that A_1 and B_1 are the same for all pixels. However, in reality, due to the rotation of unit cells, the pixel values of A_1 for unit cell number one may not be the same as the value of A_1 in unit cell two, but they are relatively close to each other. The values of A1 exclusively pertain to the pixel_1 values corresponding to conformation A. However, it is important to note that these values may not necessarily be equal.

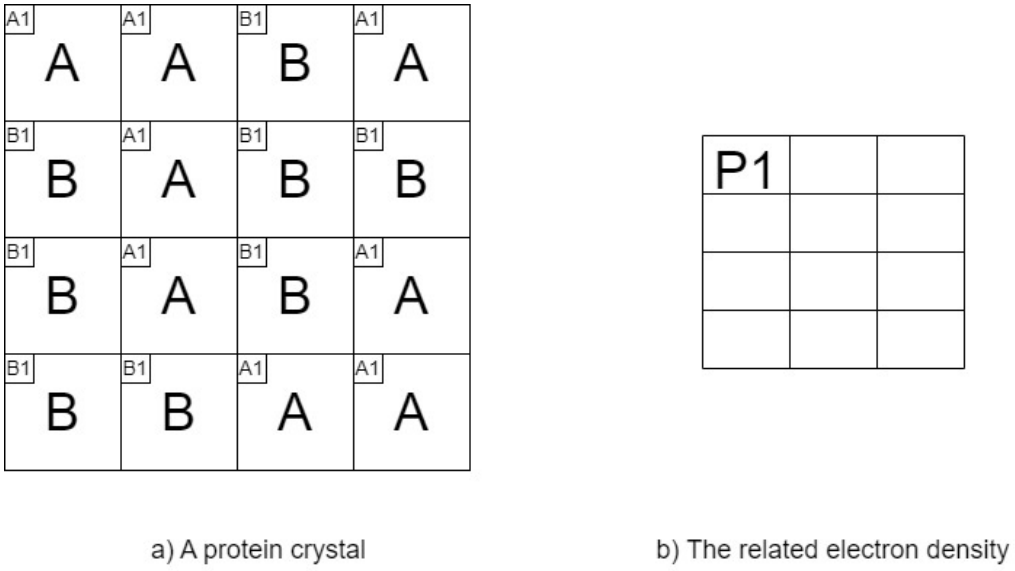


Figure 9: Schematic view of protein and electron density. Protein (a) and electron density (b)

We can extend this approach to more conformations and unit cells. This step aims to extract meaningful information from the vast numbers that contribute to the calculation of a single pixel value, such as P_1 in the previous example.

4.3.1 Mixture Model

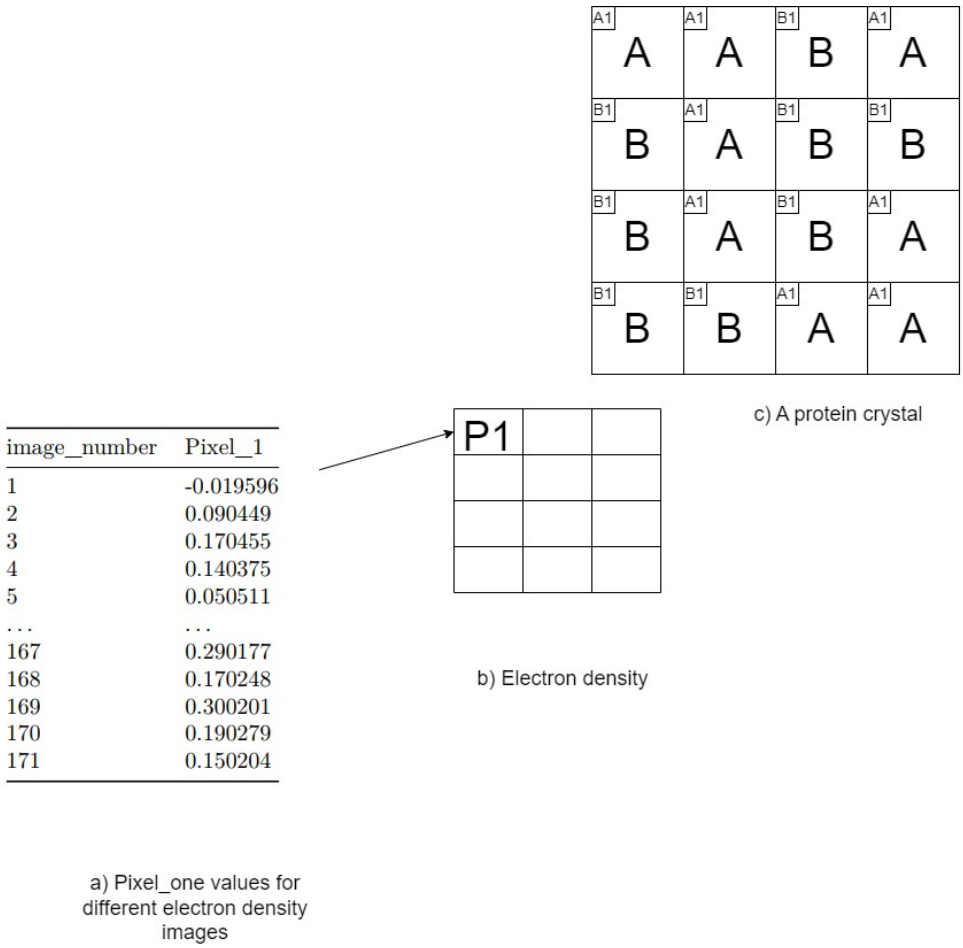
The focus will be exclusively on the pixel_1 values within the data frame, as illustrated in Figure 5. It is important to note that each value corresponds to a weighted average derived from an extensive dataset containing trillions of individual data points. We need to extract information about those original numbers to answer the research questions, as referred to throughout this thesis.

We have already become familiar with the Gaussian mixture model in Section 3.6. It is a reasonable assumption for the distribution of the original numbers(numbers which created pixel_1), and based on this assumption, each pixel in an image is a weighted average of trillions of numbers that come from a distinct Gaussian mixture model. Therefore each component in the GMM corresponds to a conformation. Figure 10 will help us better understand the concept.

P_1 value is the weighted average of all pixel one values in the protein crystal. If we denote those original values y_i, we have a probability of linear superposition of Gaussians as Equation (27).

$$p(y) = \sum_{k=1}^{M} \pi_k \cdot N(y|\mu_k, \Sigma_k) \tag{27}$$

Each Gaussian corresponds to a specific protein conformation. M represents the number of conformations, and π_k represents the proportion of each conformation in the weighted average value.



image_number	Pixel_1
1	-0.019596
2	0.090449
3	0.170455
4	0.140375
5	0.050511
...	...
167	0.290177
168	0.170248
169	0.300201
170	0.190279
171	0.150204

Figure 10: Schematic view of protein crystal and electron density values in more details

The expected value for a continuous random variable is:

$$E[y] = \int y \cdot p(y) \tag{28}$$

The expected value of y by replacing p(y) with Gaussian mixture model distribution would be:

$$E[y] = \int y \left(\sum_{k=1}^{M} \pi_k \cdot N(y|\mu_k, \Sigma_k) \right) dy \tag{29}$$

$$E[y] = \sum_{k=1}^{M} \pi_k \cdot \int y \cdot N(y|\mu_k, \Sigma_k) dy \tag{30}$$

$$\int y \cdot N(y|\mu_k, \Sigma_k) dy = \mu_k \tag{31}$$

$$E[y] = \sum_{k=1}^{M} \pi_k \cdot \mu_k \tag{32}$$

As previously stated, considering the vast number of y_i values, the expected value of the variable y, denoted as $E[y]$, can be approximated by the mean, represented as $\bar{y}$.

$$\bar{y} = \sum_{k=1}^{M} \pi_k \cdot \mu_k + \epsilon \tag{33}$$

We lack information regarding the values of π_k and μ_k, and the objective is to estimate these values in a way that minimizes the discrepancy between the predicted values $\hat{\bar{y}}$ and the actual values $\bar{y}$. $\hat{\pi}$ and $\hat{\mu}$ are the estimated values for π and μ. Here, $\bar{y}$ represents the weighted average values in the electron density map (values in the dataframe Figure 5), and $\hat{\bar{y}}$ denotes the predicted values based on the estimated values of π_k and μ_k and N is the number of data points.

The predicted values $\hat{\bar{y}}$ can be calculated using the following equation:

$$\hat{\bar{y}} = \sum_{k=1}^{M} \hat{\pi}_k \cdot \hat{\mu}_k \tag{34}$$

We can calculate the likelihood across the measurements by assuming that the error term (ϵ) follows a Gaussian distribution with a mean of zero and a fixed variance. $\bar{y_j}$ is the weighted average of each measurement. This can be expressed as:

$$\prod_{j=1}^{N} p(\bar{y_j}|\hat{\pi}, \hat{\mu}) = \prod_{j=1}^{N} \frac{1}{(\sigma\sqrt{2\pi})} \cdot e^{(-\frac{(\bar{y}_j - \hat{\bar{y}_j})^2}{2\sigma^2})} \tag{35}$$

In order to maximize the likelihood, it is necessary to minimize the error term. As discussed in Section 3.7, the MSE is a valuable metric for evaluating the disparity between the actual values $\bar{y}$ and the predicted values $\hat{\bar{y}}$. In the specific context of this study, the MSE can be computed using the following equation:

$$MSE = \frac{\sum_{j=1}^{N} (\bar{y_j} - \hat{\bar{y}_j})^2}{N} \tag{36}$$

4.3.2 Model for Two Components

Suppose we focus on a single pixel, for example, pixel_1. In that case, we have a distinct distribution for each measurement (row), and we only have a single observation from that distribution, which is the mean of the Gaussian mixture distribution. Consequently, it is not possible to determine the distribution based on a single data point. To address this issue, we need to make certain assumptions informed by expert knowledge in the domain.

Assuming that each row of data points follows a Gaussian distribution with two components, the distributions are consistent across different measurements but differ in their π values. In other words, based on this assumption, the second measurement has the same μ_1 and μ_2 as the first one, but those conformations have different proportions(π) in creating the pixel value. We can express Equation (37) for each measurement, in which π_1 is the proportion of conformation one and π_2 is the proportion of conformation two.

$$\pi_1 + \pi_2 = 1 \tag{37}$$

By considering :

$$\pi_1 = \pi \tag{38}$$

$$\pi_2 = 1 - \pi \tag{39}$$

By replacing Equations (38) and (39) in Equation (34):

$$\bar{y}_j = \pi_{1,j} \cdot \mu_1 + \pi_{2,j} \cdot \mu_2 = \pi_j \cdot \mu_1 + (1 - \pi_j) \cdot \mu_2 \tag{40}$$

In which $\pi_{1,j}$ is the proportion of conformation one in the creation of measurement of $\bar{y}_j$ and j is the number of images ranging from 1 to N=171, So:

$$\pi_j = \frac{\bar{y}_j - \mu_2}{\mu_1 - \mu_2} \tag{41}$$

In Equation (43), we can observe that the likelihood across different measurements is determined by the assumption that π follows a Dirichlet distribution and that π_j values for different values of j are independent. This can be represented as:

$$\pi_j \sim Dir(\alpha_1, \alpha_2) \tag{42}$$

Here, π_j represents the proportion of conformation one in the creation of the measurement $\bar{y}_j$, where j ranges from 1 to N=171. The parameters α_1 and α_2 correspond to the parameters of the Dirichlet distribution. It is important to note that the assumption is made that all the π values follow the same Dirichlet distribution, implying that α_1 and α_2 are constant across all measurements. This assumption allows for consistent modeling of the proportions across different conformation measurements.

$$P(\pi_1, ..., \pi_{171}) = \prod_{j=1}^{171} Dir(\pi_j, \alpha_1, \alpha_2) \tag{43}$$

The next step is taking the log of likelihood:

$$log(P(\pi_1, ..., \pi_{171})) = log(\prod_{j=1}^{171} Dir(\pi_j, \alpha_1, \alpha_2)) \tag{44}$$

By substituting the distribution function into Equation (44), we obtain:

$$log(P(\pi_1, ..., \pi_{171})) = \sum_{j=1}^{171} (log(\frac{1}{\beta(\alpha)}) + (\alpha_1 - 1) log(\frac{\bar{y}_j - \mu_2}{\mu_1 - \mu_2}) + (\alpha_2 - 1) log(\frac{\mu_1 - \bar{y}_j}{\mu_1 - \mu_2})) \tag{45}$$

So, we need to find the parameters ($\hat{\mu}$ and $\hat{\pi}$) that make $\hat{\bar{y}}$ closer to $\bar{y}$. In other words, we aim to minimize MSE in Equation (36). For this purpose, gradient descent is utilized to find the optimal parameters.

Then taking partial derivative with respect to parameters.

$$\frac{\partial log(P(\pi_1, ..., \pi_{171}))}{\partial \mu_1} = \sum_{j=1}^{171} ((\alpha_1 - 1)(\frac{-1}{\mu_1 - \mu_2}) + (\alpha_2 - 1)(\frac{1}{\mu_1 - \bar{y}_j} - \frac{1}{\mu_1 - \mu_2})) \tag{46}$$

$$\frac{\partial log(P(\pi_1, ..., \pi_{171}))}{\partial \mu_2} = \sum_{j=1}^{171} ((\alpha_1 - 1)(\frac{-1}{\bar{y}_j - \mu_2} + \frac{1}{\mu_1 - \mu_2}) - (\alpha_2 - 1)(\frac{1}{\mu_1 - \mu_2})) \tag{47}$$

$$\psi(z) = \frac{\partial \ln \gamma(z)}{\partial z} \tag{48}$$

$$\frac{\partial log(P(\pi_1, ..., \pi_{171}))}{\partial \alpha_1} = \sum_{j=1}^{171} (\psi(\alpha_1 + \alpha_2) - \psi(\alpha_1) + \log(\frac{\bar{y}_j - \mu_2}{\mu_1 - \mu_2})) \tag{49}$$

$$\frac{\partial log(P(\pi_1, ..., \pi_{171}))}{\partial \alpha_1} = \sum_{j=1}^{171} (\psi(\alpha_1 + \alpha_2) - \psi(\alpha_2) + \log(\frac{\mu_1 - \bar{y}_i}{\mu_1 - \mu_2})) \tag{50}$$

So far, the partial derivative with respect to all parameters, namely μ_1, μ_2, α_1, and α_2, is computed. However, since all optimization techniques depend on initial values, randomly chosen initial values are used to begin the optimization process. Then, after trying different learning rates, we chose a value of $1 \cdot 10^{-6}$ to balance speed and accuracy. A convergence criterion was also set to ensure that the algorithm converges.

As the algorithm runs, the MSE trend decreases overall but with occasional fluctuations. To study this, we compared the average MSE values of the first 10 iterations with the next 10 iterations after every 100 iterations. This allowed us to compare the MSE calculated for the first 10 iterations with the mean of MSE values of the 100^{th} to the 110^{th} iterations and do the same for all iterations. We chose this approach to balance computational cost while also considering that calculating MSE in each iteration would not be helpful due to fluctuations. More information on our findings will be provided in the results section.

4.3.3 General Solution

As discussed earlier, we have assumed that measurement comes from a distribution with two components. However, there may be more components, specifically, more conformations, in a protein crystal. In fact, determining the number of components is one of the research questions in this project. Indeed, if again we concentrate on pixel_1, we have 171 distributions, and each distribution has $2 \times M$ parameters, including π and μ.

As discussed earlier, we had to rely on domain expert comments to make some assumptions. One of the assumptions was that each measurement follows a Gaussian mixture model with mixing coefficients derived from a Dirichlet distribution. Another valid assumption was that all Gaussian distributions have the same mean and differ only in their respective π values. Assuming that all Gaussian mixtures share the same mean across measurements would result in a model with only one observation per distribution. However, this would not be helpful due to the many unknown parameters involved.

So, for each of the 171 distributions assumed to follow a Gaussian mixture model with M components, there are M unknown π parameters and M unknown μ parameters. Therefore, we have $171 \times M$ unknown π parameters and M unknown μ parameters. We assumed all π values are drawn from the same Dirichlet distribution to address the issue mentioned. This allows us to reduce the number of parameters we need to estimate from $171 \times M$ unknown π parameters to just M unknown α parameters.

For example, considering M=3, we would have 516 unknown parameters, including 3×171 π values and three μ values for each distribution. However, since we only have 171 $\overline{y}$ values, there is no unique solution to mathematically finding the exact values of these parameters.

Based on the central limit theorem we discussed previously, we can consider the $\overline{y}$ (sample means) to follow a normal distribution due to the large sample size. In the following formulas, the superscript of $\bar{y}$ refers to the pixel number, and the subscript refers to the image number. M denotes the number of components in all equations.

Below we see three samples of equations:

$$\overline{y}_1^{(1)} = \sum_{k=1}^{M} \pi_{k,1}^{(1)} \mu_k$$

$$\overline{y}_2^{(1)} = \sum_{k=1}^{M} \pi_{k,2}^{(1)} \mu_k$$

$$\overline{y}_3^{(1)} = \sum_{k=1}^{M} \pi_{k,3}^{(1)} \mu_k$$

4.3.4 Employing an Iterative Approach Through Metropolis Sampling

As previously mentioned, the Dirichlet distribution is helpful for modeling percentages, which applies to π values. Building on earlier assumption of two components, we further assume that π values come from a Dirichlet distribution. Therefore, instead of finding $171 \times M$, π values, we need to find the M number of α values. α is the Dirichlet distribution parameters. However, this also introduces a new challenge, as we now need to find the relationship between α and μ, which we will address in future steps.

To solve the problem of determining the optimal values of μ and α, we introduce an iterative method based on the EM algorithm. Figure 13 can be used to aid in comprehending the algorithm. Here is an overview of the method:

1. Initializing μ and α.
2. Optimizing μ using the current values of α.
3. Optimizing α using the current values of μ.
4. Iterating over steps 2 and 3 until convergence is achieved.

In what follows, we will explore each step in detail and derive the necessary equations and relationships.

- Step one: Initialization by assigning two random values to μ and α. The algorithm commences with the initialization step, which is performed only once at the beginning. During this step, α is assigned a random value between 0 and 40, while μ is assigned a random value between 0 and 10.

- Step two: Optimizing μ using the current values of α.

  This step will provide us with an optimized value for μ given the current value of α.

  Based on Equation (36) and the use of MSE as the cost function to measure the difference between $\bar{y}$ and $\hat{\bar{y}}$, the objective in this step is to minimize Equation (51).

$$\sum_{t=1}^{T} \sum_{j=1}^{N} (\overline{y}_j - \sum_{k=1}^{M} \hat{\pi}_{k,j}^{(t)} \hat{\mu_k}) \tag{51}$$

  Where N is the number of measurements, M is the number of components, and T is the number of samples generated from the Dirichlet distribution to ensure that the parameters are not dependent on random samples. In this case, a value of 50 was used for T. The goal is to find the values of μ that minimize this equation. For this purpose, the gradient descent technique is utilized to find the optimal values of μ.

- Step three: Optimizing α using the current values of μ

  The objective function remains the same as in step two; however, unlike the previous step, where it was possible to differentiate with respect to μ, differentiating with respect to α in this step is either impossible or considerably challenging. Because α represents the parameter of the Dirichlet distribution (as shown in Equation (8)), and π is a sample from this distribution. While taking the derivative with respect to π is technically feasible, it does not provide meaningful insights for the optimization process.

  For this purpose, sampling methods are employed instead of relying on gradient-based optimization methods. In this step, we generate samples from the distribution of α to facilitate the optimization process.

We start by computing the posterior distribution of α to initiate this step. Given the Bayes theorem, we have:

$$posterior \propto prior \times Likelihood \tag{52}$$

$$p(\alpha|\overline{y}, \hat{\mu}, \hat{\pi}) \propto p(\overline{y}, \hat{\pi}|\alpha, \hat{\mu})p(\alpha) \tag{53}$$

Using factorization, we obtain the following:

$$p(\overline{y}, \hat{\pi}|\alpha, \hat{\mu}) = p(\overline{y}|\hat{\pi}, \alpha, \hat{\mu})p(\hat{\pi}|\alpha, \hat{\mu}) \tag{54}$$

Since $\overline{y}$ does not depend on α given $\hat{\pi}$ and $\hat{\pi}$ doesn't depend on $\hat{\mu}$, we will have :

$$p(\overline{y}, \hat{\pi}|\alpha, \hat{\mu}) = p(\overline{y}|\hat{\pi}, \hat{\mu})p(\hat{\pi}|\alpha) \tag{55}$$

As mentioned previously, we assume that $\hat{\pi} \sim Dir(\alpha)$, which is the prior.

So:

$$p(\overline{y}, \hat{\pi}|\alpha, \hat{\mu}) = \frac{1}{\sqrt{2\pi}} e^{\frac{-1}{2}(\overline{y} - \sum_{k=1}^{M} \hat{\pi}_k \hat{\mu}_k)^2} \frac{1}{\beta(\alpha)} \prod_{k=1}^{M} \hat{\pi}_k^{\alpha_k - 1} \tag{56}$$

In Equation (56), Note that we need to make a distinction between $\hat{\pi}_k$ and the fixed value of π, which equals 3.14.

Likelihood is:

$$\prod_{j=1}^{N} p(\overline{y}_j, \hat{\pi}|\alpha, \hat{\mu}) = \prod_{j=1}^{N} \left(\frac{1}{\sqrt{2\pi}} e^{\frac{-1}{2}(\overline{y} - \sum_{k=1}^{M} \hat{\pi}_k \hat{\mu}_k)^2} \frac{1}{\beta(\alpha)} \prod_{k=1}^{M} \hat{\pi}_k^{\alpha_k - 1} \right) \tag{57}$$

The choice of the prior distribution allows for flexibility, as any distribution with values greater than zero can be selected. In this study, the exponential distribution with a parameter $\lambda = 0.1$ was chosen as the prior distribution. However, the selection of λ the expected values for α. By comparing two different values of λ for the exponential distribution, we can assess the impact on the tail ends and the distribution of samples. In Figure 11, with $\lambda = 0.1$, the tail ends around 40, while in Figure 12, most samples are concentrated between 0 and 0.25. The choice of λ aims to provide a less informative prior that does not constrain α values to be around zero. However, the suitability of this choice can be further evaluated by examining the resulting α values obtained from running the model. Suppose the optimal α values are predominantly located around the tail of the distribution (40). In that case, it suggests that a lower value of λ for the distribution of α would have been more appropriate as the prior.

So:

$$\alpha \sim exp(0.1) \rightarrow p(\alpha) = \lambda e^{-\lambda \alpha} = 0.1 e^{-0.1\alpha} \tag{58}$$

By considering joint distribution of α values have identical and independent distributions(iid):

$$p(\alpha_1, \alpha_2, ..., \alpha_M) = (0.1)^M \prod_{k=1}^{M} e^{-0.1\hat{\pi}} \tag{59}$$

We can rewrite the posterior as :

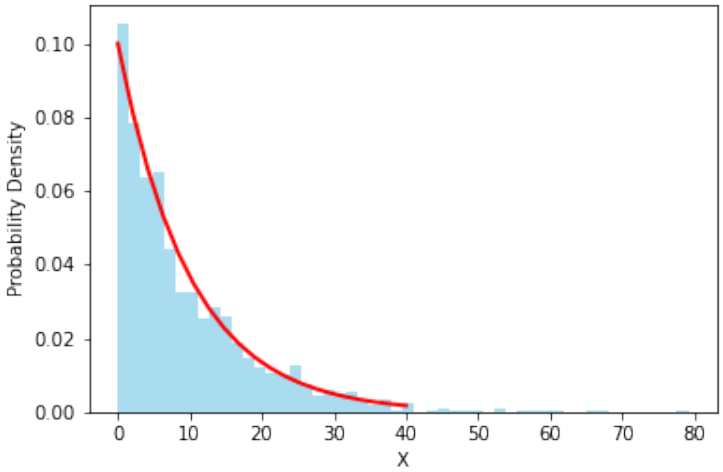


Figure 11: Prior exponential distribution of samples with $\lambda = 0.1$

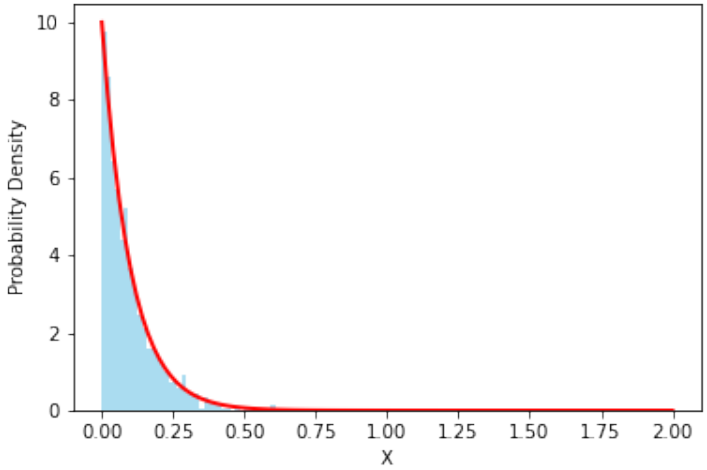


Figure 12: Prior exponential distribution of samples with $\lambda = 10$

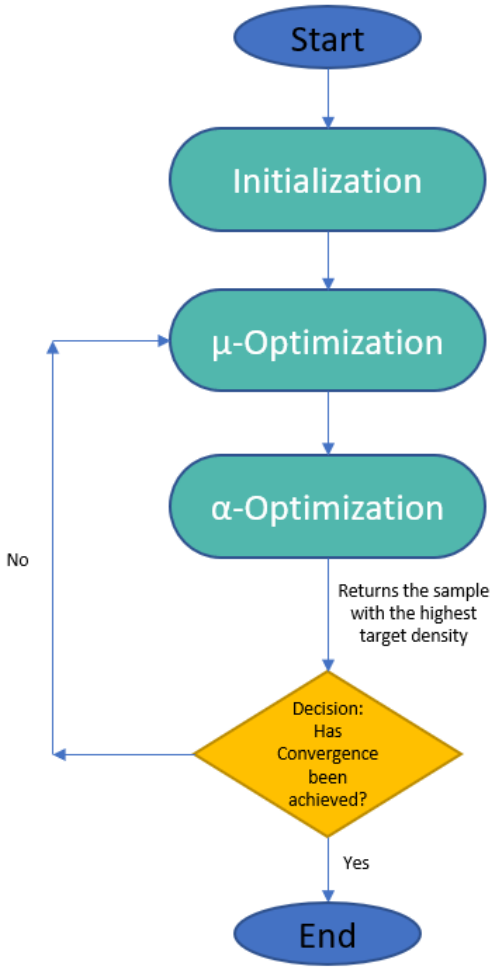


Figure 13: Algorithm for generalized solution

$$p(\alpha|\overline{y},\hat{\mu},\hat{\pi}) \propto \frac{N}{\sqrt{2\pi}} e^{\frac{-N}{2}(\overline{y}_j - \sum_{k=1}^{M} \hat{\pi}_{k,j}\hat{\mu}_k)^2} \frac{1}{\beta(\alpha)^N} (\prod_{j=1}^{N}\prod_{k=1}^{M} \hat{\pi}_{k,j}^{\alpha_k - 1}) \cdot e^{-0.1\sum_{k=1}^{M}\alpha_k} \quad (60)$$

Since calculating the partial derivative with respect to α in the objective function (Equation (51)) is challenging due to its nature as a parameter of the Dirichlet distribution, the Metropolis-Hastings algorithm is employed to sample values of α from the target distribution.

In this step, we utilize the Metropolis-Hastings algorithm to generate many samples from the target distribution, $p(\alpha|\overline{y},\mu,\pi)$. After generating the samples, we select the sample with the highest density value and update the initial value of α.

The normal distribution is used as the proposal distribution for the Metropolis-Hastings algorithm. The returned α value is then given to step 2, and we iterate over steps 2 and 3 until a convergence criterion is satisfied. The overall form of the algorithm is depicted in the Figure 13.

When implementing the algorithm, it was observed that the posterior probability could sometimes tend toward positive or negative infinity. To ensure numerical stability, the logarithm of the posterior probability is taken (Equation (61)). Optimizing the logarithmic function is equivalent to optimizing the original posterior probability since

it does not alter the order of numbers. In order to invert the logarithm and obtain the acceptance ratio in the Metropolis algorithm, the exponential function is applied to the logarithmic value. This allows for proper calculation and comparison of acceptance probabilities during sampling.

$$log(p(\alpha|\overline{y}, \hat{\mu}, \hat{\pi})) \propto \frac{-N}{2} \sum_{j=1}^{N} (\overline{y}_j - \sum_{k=1}^{M} \hat{\pi}_{k,j} \hat{\mu}_k)^2 + \sum_{j=1}^{N} log(Dir(\alpha)) - 0.1\alpha \tag{61}$$

5 Results

This section will present the results of applying different methods described in Section 4 to the data. We will start by presenting the results obtained from applying PCA. Next, we will present the outcomes obtained by running different clustering algorithms. Then, using the clustering results, we will partition the data into training and testing sets and utilize them to train both two-component and general models.

5.1 PCA

As mentioned earlier, the high dimensionality of data, which includes 432000 features(pixels), makes it challenging to cluster the images effectively. To address this issue, we performed PCA with 11 and 31 components, which could explain 80% and approximately 90% of the data variation, respectively. This dimensionality reduction allowed us to obtain three sets of data: the original data, data with 11 components, and data with 31 components. These data sets will be only used for clustering in the following steps to obtain more reliable results.

5.2 Clustering

To determine the optimal model for outlier removal, we employ three distinct clustering methods: k-means, DBSCAN, and GMM. Subsequently, we compare the outcomes of these methods on the three aforementioned datasets.

5.2.1 K-means

Initially, the performance of k-means clustering was assessed using all three datasets. The objective was to identify and remove outliers in order to group the most similar measurements. Therefore, we chose k=2 as two clusters were enough for this purpose.

After clustering, it was found that there were 171 measurements in Cluster 1 and 55 measurements in Cluster 2, and this result was consistent across all datasets, including the original, 11-component, and 31-component datasets. It appears that the noisy data points were effectively removed. The result of the k-means clustering is shown in Figure 14, where PC1 and PC2 are the dimensions.

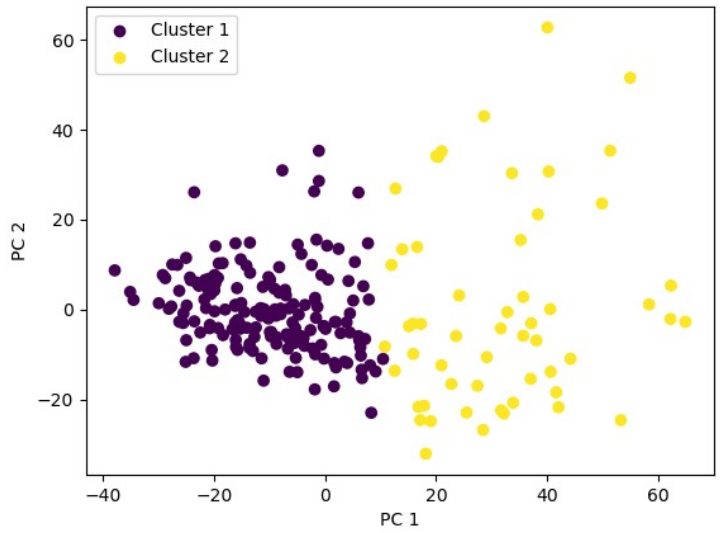


Figure 14: The result of k-means clustering

5.2.2 DBSCAN

DBSCAN was chosen as the second clustering method. However, the results were consistent across all datasets, and the model did not perform well. Adjusting the parameters manually posed challenges in achieving optimal clustering outcomes, with some resulting in oversimplified clusters and others in overly complex clusters. Figure 15 illustrates an example of the results. The image on the right shows the result obtained when setting min-sample=1. It is worth noting that the assigned labels are arbitrary and do not possess any intrinsic significance. In the left image, when the min-sample parameter was modified from 1 to 2, all data points were grouped into a single cluster.

5.2.3 GMM

GMM is a distribution-based clustering algorithm. In GMM, each Gaussian distribution is considered as a cluster. It is a soft clustering method.

Below, we can see the result of the model on three different datasets.

Initially, we attempted to perform clustering using GMM on the original dataset. However, we encountered a memory issue during the processing of the dataset.

169 data points were in one cluster and 57 data points in another cluster, but some data points that were similar to each other were classified in different clusters(Figure 16).

The same outcome as the k-means clustering method was obtained using 31 components. Based on the improved classification results depicted in Figure 17, we decided to proceed with this outcome. Consequently, 55 measurements were removed, resulting in a dataset consisting of 171 data points (Figure 17).

Figure 18 illustrates the histogram of all the measurements and those grouped in Cluster 1.

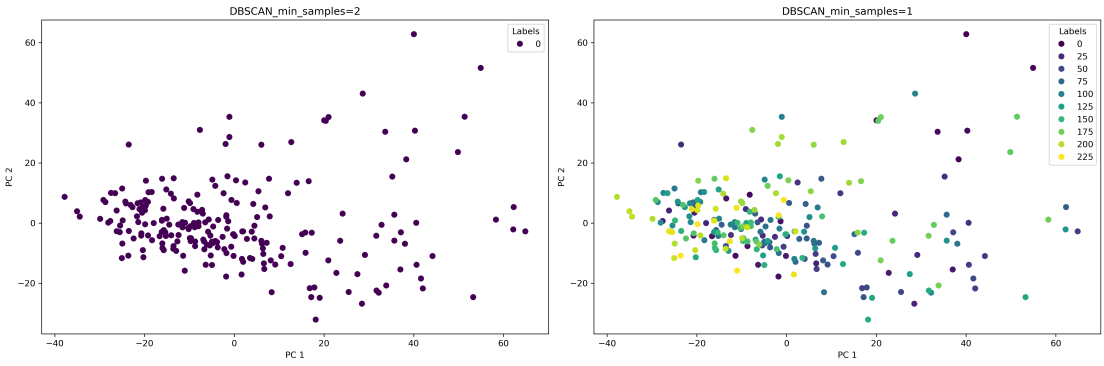


Figure 15: DBSCAN min samples=1 and 2

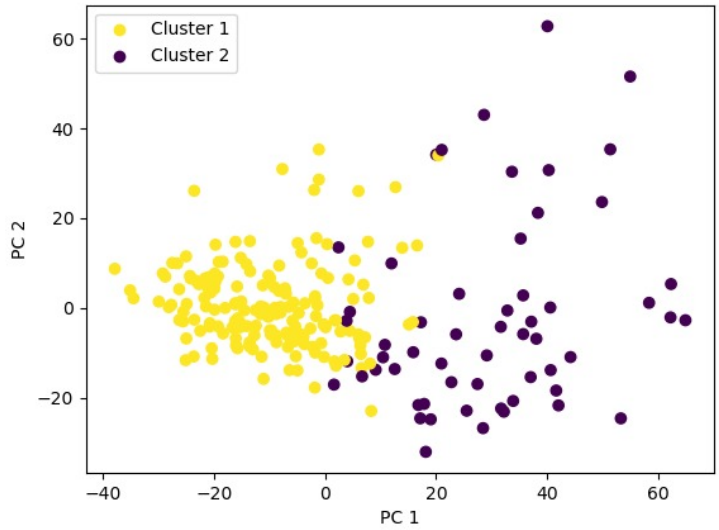


Figure 16: The outcome of applying GMM to the dataset with 11 components

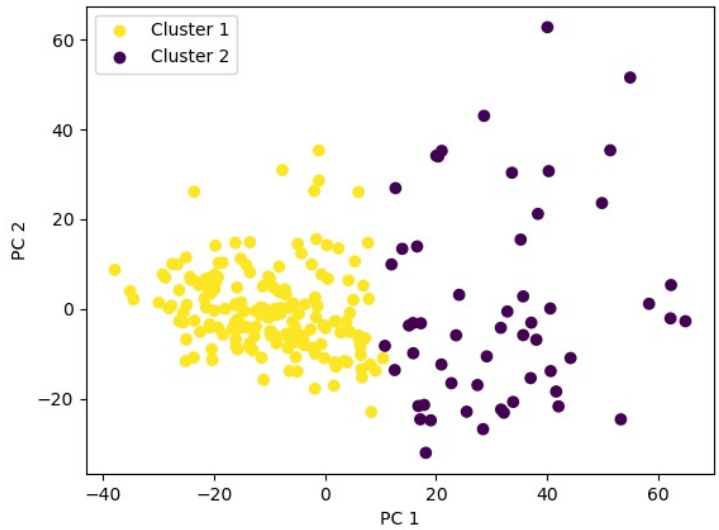


Figure 17: The outcome of applying GMM to the dataset with 31 components

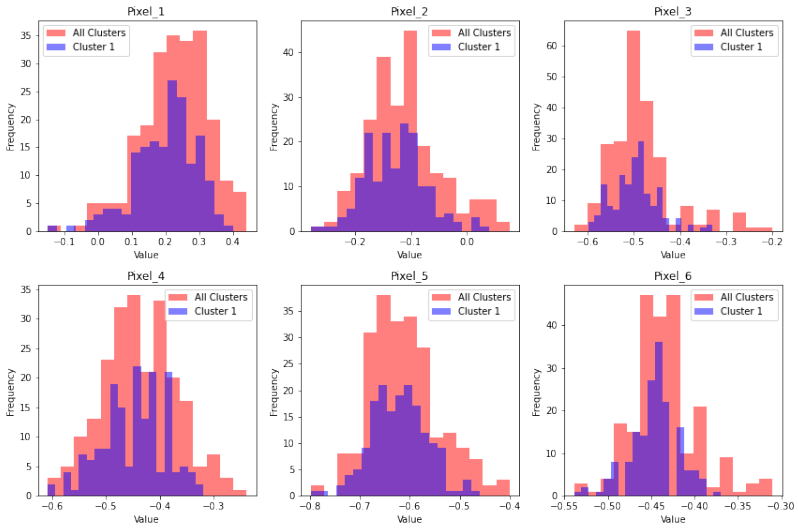


Figure 18: Histogram of the first six pixels demonstrating a normal distribution pattern(all data points versus Cluster 1)

5.3 Data partitioning

In the future, when applying the model, we will need two separate data sets for training and testing. We split the data into a training dataset (containing 70% of the data) and a test dataset (containing the remaining 30%). This resulted in a training dataset containing 119 data points and a test dataset containing 52 data points.

5.4 Model for Two Components

As previously demonstrated, the initial algorithm was capable of solving the problem for two components but was not effective in generalizing the model for M components. Therefore, a new algorithm was used to estimate the parameters of the general model. In this section, we will first analyze the results of the model with two components and then examine the results of the generalized model in the hope of discovering interesting findings.

As mentioned earlier, the proposed model is achieved by assuming the original numbers are drawn from a Gaussian mixture model with two components, where the mixing coefficients of the Gaussian mixture are sampled from a Dirichlet distribution. We implemented this model on pixel_1, and then we will compare the results with the generalized solution to help us understand the efficacy of the model and recognize the limitations of considering M=2 for all pixels, which is far from reality.

Firstly, we needed to find the optimal learning rate value. A set of learning rates was selected, and the resultant MSE values were compared as drawn in Figure 19. The optimal learning rate was chosen at the elbow point, which was found to be 1e-6.

We used the optimal learning rate to run the model on pixel_1. The model was trained on 119 measurements of training data and tested on the test data, which includes 52 measurements. Figure 20 depicts the trend of the MSE training plot over more than 21000 iterations, and its convergence is apparent in the plot. Table 1 displays the model results and the final obtained values for μ and α.

Table 1: MSE result for two component model

μ_1	μ_2	α_1	α_2	$MSE-train$	$MSE-test$
2.19	-0.89	1.08	1.55	0.668	0.671

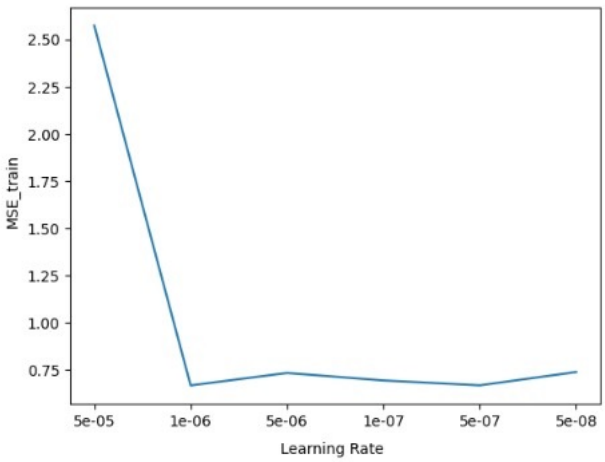


Figure 19: Trend of training MSE for some learning rates

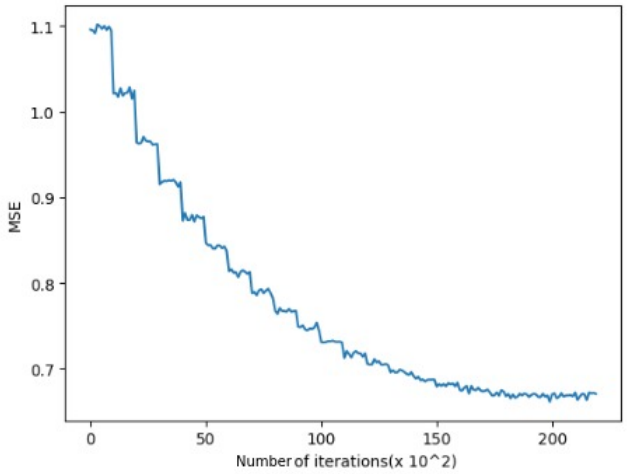


Figure 20: Trend of training MSE over 21000 iterations

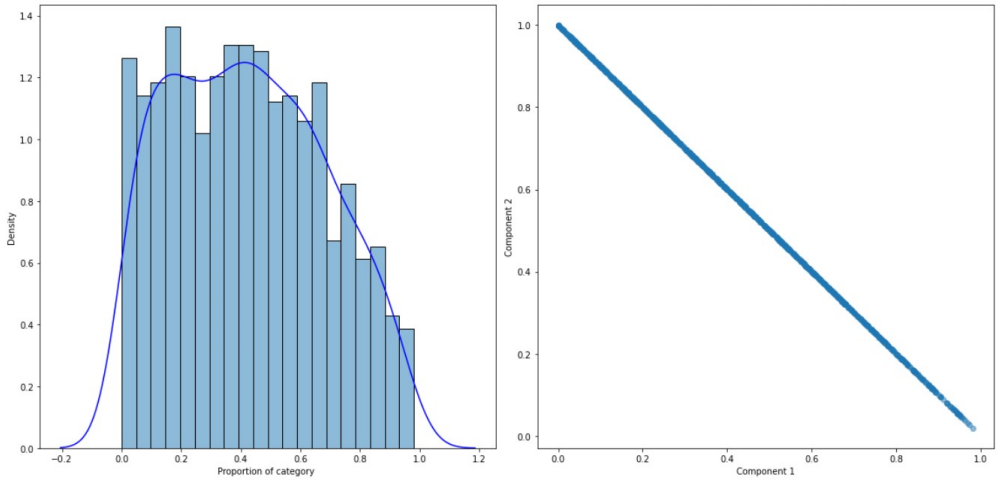


Figure 21: Dirichlet samples of obtained parameters. It shows the distribution of π obtained from the α values. This indicates the proportion of each component in the mixture, where values close to 0 or 1 indicate that the pixel belongs to one of the two Gaussian components. Pixels with intermediate values, which make up the majority of the distribution, suggest that both components have almost equal proportions in creating the pixel of electron density. Further investigation of these pixels could provide insight into the nature of their distribution and give us information about the overall protein structure.

5.5 Generalizing Algorithm for Different M

As discussed before, the previous model was unable to generalize for M components. In this subsection, we will first determine optimal step size values and then evaluate the results of the generalized model on pixel_1 for M=2 components, comparing them with the previous model's results. We will then run the model for ten different pixels to determine α values in the hope of getting interesting results.

5.5.1 Strategy for Selecting the Optimal Step Size

For the choice of step size, we iterate over a set of values, calculate the number of accepted samples, and then choose which one fits more on our purpose according to the results.

We conducted an iterative process to select the optimal step size by calculating the number of accepted samples for each value. This process was repeated four times, with 50 samples drawn from the Metropolis in step 3 of the general algorithm for each iteration. The results of this analysis are presented in Figure 22.

According to Gelman and Roberts [31] and Roberts and Rosenthal [32], the optimal acceptance rate is approximately 0.234. By observing Figure 22, we can see that the step size of 2, which resulted in around 20 accepted samples out of 200, is closer to the optimal acceptance rate. Therefore, based on this criterion, a step size of 2 has been selected.

5.5.2 Modeling for Two Conformations

In this step, we implemented the model for M=2 and used the previously obtained step size for pixel_1. As previously discussed, we calculated the MSE train for the first five samples in every ten iterations as the convergence criterion because the MSE trend fluctuates. Additionally, calculating the MSE based on α in each iteration is time-consuming and computationally expensive. It is worth noting that for this implementation, the number of Metropolis samples in step 3 and the number of generated samples from the Dirichlet distribution were set to 50.

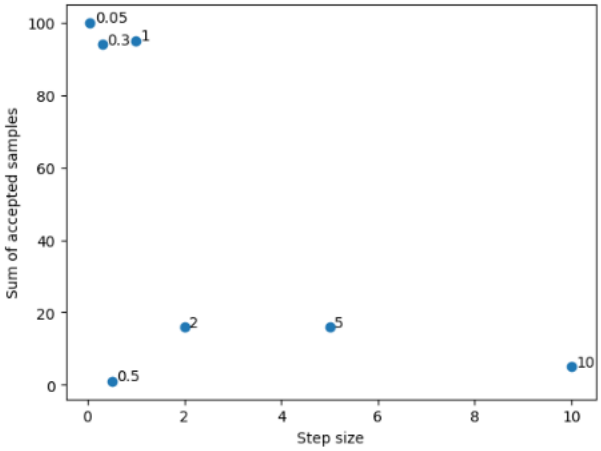


Figure 22: Comparison of step size and accepted samples

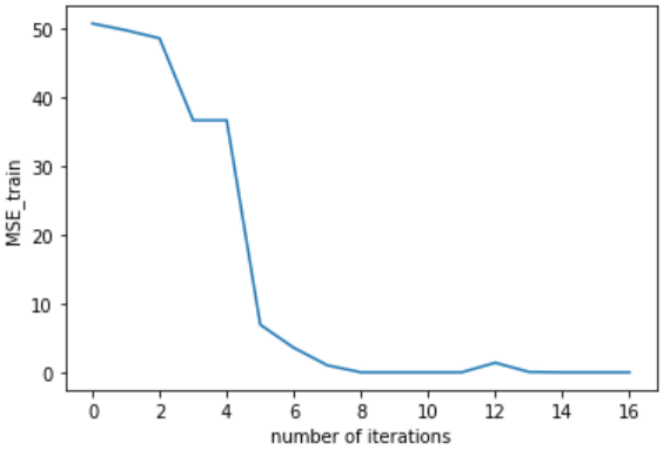


Figure 23: The trend of the training MSE over different iterations, which converges after 31 iterations.

By comparing the results, we can observe that the generalized model works better for pixel_1. Although maybe M=2 is not its optimal M. In Table 2, we can see the result of running a generalized model on pixel_1 for M=2.

Figure 24 shows the distribution of π based on the resulting α. As we can see, the values of one category are mostly higher than the other.

Table 2: MSE result for generalized model

μ_1	μ_2	α_1	α_2	$MSE - train$	$MSE - test$
0.26	6.33	8.87	1.61	0.013	0.007

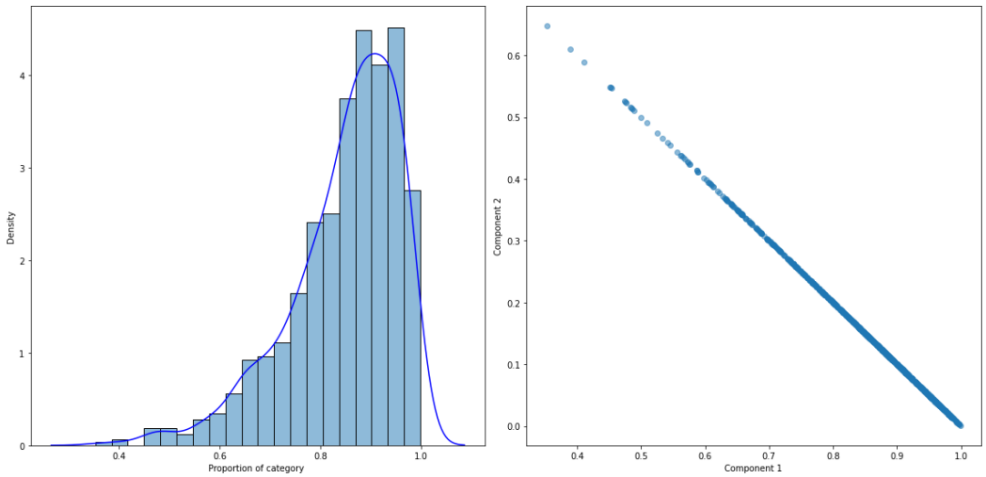


Figure 24: Dirichlet samples of obtained α

5.5.3 Running the Model for 10 Pixels

In this step, we ran the model over 10 randomly chosen pixels to compare the results and obtain reasonable values. We selected 2 random positions and chose 5 pixels from each site to demonstrate the similarity in α values.

If we look back at Figure 8, we will notice that the nearest pixels have more similarities than the far pixels. This indicates the continuity of the image and not seeing any sharp change in the image, as the nearest pixels form almost the same shape. Therefore, we can expect similar results in the nearest pixels, or at least in highly correlated ones.

The resultant values of α were expected to be very similar, and it is necessary to sort the α since they are not in the same order. The pixels do not have to be different, but those correlated (closest pixels) are expected to be similar.

Initially, we applied the model over the pixels to find the optimal number of components. Then we ran the model on each pixel using the optimal M and returned the resultant α and μ. In order to select an optimal value for M, we have utilized an approach whereby this value corresponds to the point at which the MSE trend begins to plateau. For other instances where the MSE shows fluctuation, the minimum value has been chosen

The pixel numbers from 40000 to 40004 and 430000 to 430004 were chosen for comparison. As we see in Figure 25, the optimal M for most of the pixels were 4,5, and 6.

The next step involves sorting the α values and adding 0 for cases with fewer α values than pixels according to their respective optimal M. The resulting α values were sorted and then visualized using the PCA method with two components to facilitate the visualization of the results.

The expectation was to obtain α values that are similar. However, it is noticeable that most of the points are not so close to each other. In this set of pixels, pixels 430003 and

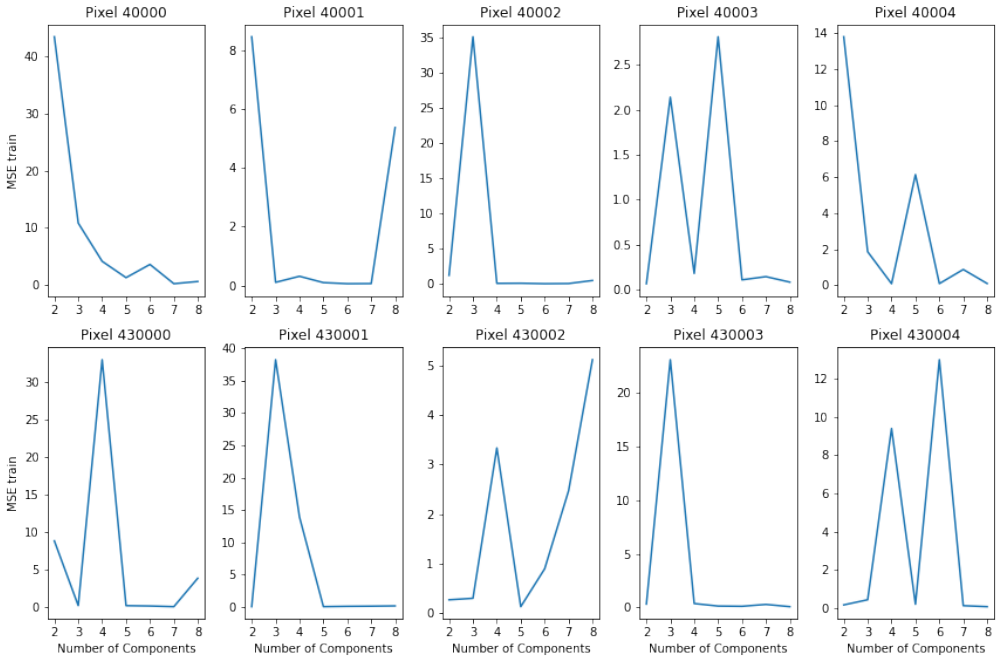


Figure 25: MSE trend for 10 pixels

40002 are notably close to each other. The interesting aspect of this similarity is that they have a similar trend in terms of MSE and an optimal M value of 4. We plotted α values in Figure 26 to see how similar or different they are.

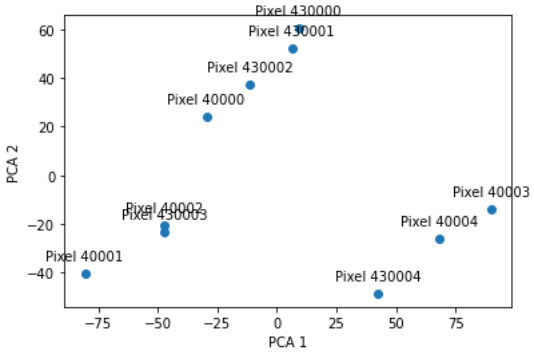


Figure 26: Scatter plot representing the similarities of α values

6 Discussion

This section commences with a discussion of the responses to the research questions, followed by the investigation of the assumptions made during the research process. Thereafter, deliberation of the study's constraints will ensue. The methodology entailing the application of the algorithm across all pixels will also be elucidated. The discussion will then broaden to encompass a specific research question that remained unresolved due to time constraints during the project's tenure. Lastly, potential avenues for future research building upon this work will be suggested.

6.1 Research Questions

Starting with the first research question, "Can multi-dataset analysis reveal how many sub-states are present in each region?" The answer to this question is related to the optimal M value. With the generalized model, we can find the optimal M value for each pixel, which reflects the number of conformations involved in creating that region. Therefore, the model can answer this research question with a "yes".

The second research question is, "Can multi-dataset analysis reveal which sub-states are present in each region?" Again, the answer from our model for this question is also yes. Using the π values, we can find which sub-states (conformations) and their proportion in forming each region.

The third research question we have answered with the generalized model is, "Which map regions are most variable, and how does this correlate to structural variability?" Using the generalized model, we can find the optimal M for each pixel by running the model on some sites of the image and calculating the optimal M based on the MSE trend. Those regions(neighbor pixels) with higher differences in optimal M values are more variable than the others, meaning they are created from more conformations. This also means that these pixels are a weighted average of more conformations than the other pixels. On the other hand, as previously mentioned, the nearest pixels exhibit a high degree of correlation. Therefore, we would expect to obtain nearly identical values within a given region, and we do not anticipate any significant sharp changes in the optimal M across the neighbors in the electron density image.

6.2 Investigating the Assumptions

The accomplishments presented in this thesis are contingent upon several reasonable assumptions.

The initial assumption asserts that all 171 electron density images have been created by the same conformations but different proportions. This presumption is justified considering the previous clustering of images based on similarities and corroborated by domain experts who state that fewer than 10 distinct conformation types are observed in real-world scenarios.

A further assumption pertains to the pixel values corresponding to a specific conformation type, which are hypothesized to adhere to a Gaussian distribution. This conjecture is considered suitable based on the understanding that pixel values for an identical conformation, while not precisely equivalent, exhibit significant similarity.

Additionally, the study operates under the assumption that the mixing coefficients in the Gaussian mixture model adhere to a Dirichlet distribution. This assumption was selected as it reduces the number of unknown parameters, and the Dirichlet distribution serves as an appropriate choice for modeling percentages.

6.3 Limitations

The results presented above are based on the Metropolis-Hastings algorithm with 50 samples and 50 Dirichlet samples. Due to time limitations, higher samples and iterations were not explored, but more accurate results are expected to be obtained with increased

sampling and iterations. In step 3 of the proposed algorithm, some of the proposed values may be less than zero. In those cases, we generate a new proposal. However, if a significant number of proposals are negative, this process can be time-consuming. Another limitation in step 3 is that overflow issues can occur for some proposal samples, which requires us to generate a new proposal for α. If this happens frequently, it can become time-consuming.

Like all optimization techniques, the model is sensitive to initial values, and altering the initial values can lead to different parameter estimates.

When choosing the step size in the generalized model, a single run of the model is insufficient. It is advisable to execute the model multiple times and subsequently compute the mean of the accepted samples to reach a more accurate result.

In specific iterations of the experiment, it was observed that the MSE for the training and test datasets were very similar, but the MSE for the test dataset was slightly lower than the MSE for the training dataset in some cases. This phenomenon may be attributed to the small size of the dataset. It could potentially be mitigated by increasing the number of samples generated from the Dirichlet distribution (T in Equation (51)).

Because of the novelty of this research, we did not have any other articles that had specifically done this task before, making it difficult to compare the results with previous studies.

6.4 Extension of the Algorithm Across Pixels

The model was initially developed for one pixel and then repeated across 10 pixels from two different sites of the image. We can generalize the model to all pixels. However, optimizing all pixels at once might be a more challenging task. The basic idea remains the same, with the equation being almost identical, except we would need to minimize the sum of the equation over different dimensions (pixels). In the current equation, we have summed over the dataset(measurements). However, we can add an additional sum over pixel r to optimize over all pixels at once. We need to minimize the equations below to run the algorithm on all the pixels, which will help us obtain optimized values for all pixels.

$$\sum_{r=1}^{R}\sum_{j=1}^{N}\sum_{t=1}^{T}(\overline{y}_{j,r} - \sum_{k=1}^{M}\pi_{k,j}^{(t)}\mu_{k,r}) \tag{62}$$

In Equation (62), R represents the number of pixels, which is 432000, and it should be noted that both $\overline{y}$ and μ vary across the pixels. In this situation, we would have a 2-dimensional array of μ instead of a list of μ, as each pixel has its own set of μ. Similarly, we expect to have a 2-dimensional array of $\overline{y}$ for all pixels. We can share the π across each measurement (image) since we know the α are the same across the pixels. We also need to use all the pixels in step 2 and step 3. Hence, we need to minimize the above function in step 2.

In the generalized approach on the result section (Section 5.5), we explained that the α values must be identical across the pixels. However, we attempted to run it but could not complete it due to computational limitations and time constraints.

6.5 Long-distance Correlations

One question we did not have enough time to address is the following research question:

- Can long-distance correlations be detected between different regions?

Answering this question requires finding dependencies among pixels which we could not explore further due to time limitations.

6.6 Future Work

The results obtained within this thesis are preliminary, further investigation is needed to explore this subject.

As mentioned before, running the algorithm on a higher number of Metropolis and also Dirichlet samples could provide better results.

Having only 226 images for training, the model was insufficient to obtain good accuracy. A more extensive dataset could result in more accurate training and testing results. Additionally, with a larger dataset, we could divide the data into three parts and perform hyperparameter tuning on the validation set. However, obtaining a more extensive dataset in this field is challenging, as we discussed with domain experts.

The resulting α values may not be very close together initially due to the wide range of values chosen for μ and α at the start of the algorithm. However, after evaluating the results, it was determined that using a narrower range for both μ and α yields better values. For example, starting μ with a random value between 0 and 1 based on the observed measurement values can be effective. On the other hand, starting α with the same initial values, such as (1, 1, 1, 1) for four components, is recommended. It is important to note that starting μ with the same values is not recommended, as it may require multiple iterations to update properly, as observed in the experiment.

7 Conclusion

In conclusion, this thesis aimed to extract information about the three-dimensional architecture of macromolecules using diffraction pattern images. The dataset utilized in this research consisted of 226 diffraction pattern images. Data were filtered to include only the spots present in all images, utilizing their reciprocal dimensions. The images were transformed into electron density maps through Fourier transform, providing valuable insights into the positions of atoms within the protein structure. An alignment technique was employed to ensure accurate analysis of the pixel values, followed by data flattening to create a standardized data frame for further analysis.

To group similar measurements, a clustering method was employed, supported by principal component analysis to reduce the dimensionality of the data. Additionally, a Gaussian mixture model was used to model the pixel values of different conformations, with statistical analysis performed using both a two-component model and a more generalized model with M components. An iterative algorithm was implemented to optimize the parameters of the generalized model.

Comparing the MSE values between the two-component model and the generalized model revealed that the latter performed better in capturing the underlying characteristics of the data. The results indicated that some pixels exhibited similar values for M and α, while others showed distinguishable patterns.

The proposed generalized model demonstrated its ability to determine the number of conformations involved in generating each pixel. The number of components in the Gaussian mixture model was considered as the number of conformations present.

The mixing coefficient of the Gaussian mixture model was utilized to estimate the proportion of each conformation contributing to the creation of a particular pixel. Higher proportions indicate a greater impact of the corresponding conformation in generating the pixel value.

Furthermore, the model successfully identified regions within the protein crystal with the highest variability. While the analysis focused on specific pixels, running the model on neighboring pixels is expected to reveal a gradual change in the number of conformations.

Overall, this research has demonstrated the potential of statistical analysis and machine learning models in extracting valuable information about the three-dimensional structure of proteins, which could enhance our understanding of life science and biology and aid in advancing novel pharmaceuticals. The findings highlight the potential for extracting valuable insights from complex diffraction pattern images and offer new avenues for further exploration in the field of macromolecular analysis.

8 Ethical Considerations

The following ethical considerations were taken into account throughout the research process:

1. Informed consent: The protein data used in this study were obtained from publicly available sources [4]. The data had already been collected and made publicly accessible with appropriate consent and ethical approval. Therefore, no additional informed consent was required for this specific research. The ethical considerations and protocols governing the collection of this data were already in place at the time of its original acquisition.
2. Data privacy and confidentiality: It is important to note that this research does not involve any human data. The study solely focuses on protein data, and therefore, there are no privacy or confidentiality concerns related to individuals' personal or sensitive information.

[4] https://zenodo.org/record/48770#.ZG1SQnZBy5c

References

[1] H. Rangwala and G. Karypis, *Introduction to protein structure prediction: Methods and algorithms*. John Wiley & Sons, 2011.

[2] B. Rupp, *Biomolecular crystallography: Principles, practice, and application to structural biology*. Garland Science, 2009.

[3] K. R. Acharya and M. D. Lloyd, "The advantages and limitations of protein crystal structures," *Trends in pharmacological sciences*, vol. 26, no. 1, pp. 10–14, 2005.

[4] N. Chaffey, B. Alberts, A. Johnson, *et al.*, *Molecular biology of the cell. 4th edn.* Oxford University Press, 2003.

[5] F. Stellato, D. Oberthür, M. Liang, *et al.*, "Room-temperature macromolecular serial crystallography using synchrotron radiation," *IUCrJ*, vol. 1, no. 4, pp. 204–212, 2014.

[6] A. Souza, L. B. Oliveira, S. Hollatz, *et al.*, "DeepFreak: Learning crystallography diffraction patterns with automated machine learning," *arXiv preprint arXiv:1904.11834*, 2019.

[7] M. Vollmar and G. Evans, "Machine learning applications in macromolecular X-ray crystallography," *Crystallography Reviews*, vol. 27, no. 2, pp. 54–101, 2021.

[8] V. Venkatraman and P. A. Carvalho, "On the value of popular crystallographic databases for machine learning prediction of space groups," *Acta Materialia*, vol. 240, p. 118353, 2022.

[9] M. V. Shapovalov and R. L. Dunbrack Jr., "Statistical and conformational analysis of the electron density of protein side chains," *Proteins: Structure, Function, and Bioinformatics*, vol. 66, no. 2, pp. 279–303, 2007.

[10] P. Emsley, B. Lohkamp, W. G. Scott, *et al.*, "Features and development of Coot," *Acta Crystallographica Section D: Biological Crystallography*, vol. 66, no. 4, pp. 486–501, 2010.

[11] J. W. Cooley, P. A. Lewis, and P. D. Welch, "The fast Fourier transform and its applications," *IEEE Transactions on Education*, vol. 12, no. 1, pp. 27–34, 1969.

[12] M. S. Goorsky, *X-ray and electron diffraction for epitaxial structures*. North-Holland, 2015.

[13] A. Ali, Y. W. Chiang, and R. M. Santos, "X-ray diffraction techniques for mineral characterization: A review for engineers of the fundamentals, applications, and research directions," *Minerals*, vol. 12, no. 2, p. 205, 2022.

[14] C. Kittel and P. McEuen, *Introduction to solid state physics*. John Wiley & Sons, 2018.

[15] E. O. Brigham, *The fast Fourier transform and its applications*. Prentice-Hall, Inc., 1988.

[16] F. R. Mulla and A. K. Gupta, "A review paper on dimensionality reduction techniques," *Journal of Pharmaceutical Negative Results*, vol. 13, no. 3, pp. 1263–1272, 2022.

[17] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," *arXiv preprint arXiv:1403.2877*, 2014.

[18] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[19] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[20] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[21] S. Lloyd, "Least squares quantization in PCM," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[22] G. A. Wilkin and X. Huang, "K-means clustering algorithms: Implementation and comparison," in *Second international multi-symposiums on computer and computational sciences (IMSCCS 2007)*, IEEE, 2007, pp. 133–136.

[23] M. Ester, H.-P. Kriegel, J. Sander, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Second international conference on knowledge discovery and data mining (KDD)*, 1996, pp. 226–231.

[24] M. Hahsler, M. Piekenbrock, and D. Doran, "Dbscan: Fast density-based clustering with R," *Journal of Statistical Software*, vol. 91, pp. 1–30, 2019.

[25] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *The annals of statistics*, vol. 1, no. 2, pp. 209–230, 1973.

[26] D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical statistics with applications*. Cengage Learning, 2014.

[27] S. Reid, "What is so normal about the normal distribution?" *BMJ Ment Health*, vol. 13, no. 4, pp. 100–100, 2010.

[28] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[29] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, *et al.*, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[30] C. Andrieu and J. Thoms, "A tutorial on adaptive MCMC," *Statistics and computing*, vol. 18, pp. 343–373, 2008.

[31] A. Gelman, G. Roberts, and W. Gilks, "Efficient Metropolis jumping hules," *Bayesian statistics*, 1996.

[32] G. O. Roberts and J. S. Rosenthal, "Optimal scaling for various Metropolis-Hastings algorithms," *Statistical science*, vol. 16, no. 4, pp. 351–367, 2001.