Quantifying Genetic Constraint in Protein Regions

Dharani Palanisamy (z5260276)
Faiyam Islam (z5258151)
Hilary Cao (z5308506)
Kelly Tao (z5259876)
Pooja Saianand (z5312416)

Contents

Mission Statement

Specialising in the application of machine learning and data modelling techniques for the analysis of complex biological data, Protolytix delivers world-class research in the understanding of proteins, one of the most important molecules for the future of humanity.

1. Executive Summary

To be able to detect, diagnose and prevent chronic diseases before they have had a chance to impact a person's life would have a profound impact on many around the world; however, current metrics surrounding disease prevention are vague and lack statistical measurements. To combat the ambiguous metrics surrounding current disease prevention tracking, this report aims to define, identify and assess the statistical significance of genetic constraints through the quantifiable distance between amino acids, which are the building blocks of protein. A multi-stage approach was taken; in stage zero, the given dataset was first explored through exploratory data analysis. Amino acids were then grouped together based on proximity in stage one using clustering techniques, and stage two involved evaluation of those clustering techniques. Evaluation using our chosen metrics in stage two show that K-Means clustering is the most promising model given its ability to minimise the average gamma value. Overall, Protolytix' approach to the quantification of genetic constraints using the distance between amino acids in proteins appears to be promising, however most certainly requires further study with larger and more varied samples.

2. Background

Proteins are a major component of all living systems, ranging from bacteria and viruses through to mammals such as humans (Whitford, 2013). They are responsible for the function of nearly every task in any organism's cellular machinery, and as such, are essential to life. To provide just a few examples, in humans, proteins transmit information from DNA to RNA, synthesise new molecules, and are also responsible for the formation of antibodies as part of our body's immune response (Keskin et al., 2008). One of the most critical functions of proteins is to catalyse the thousands of biochemical reactions in the human body. Amino acids are synthesised through this reaction (Fisher, 1985), and they form the building blocks of protein, which are then the building blocks of life. With trillions of mutations occurring every day in our bodies (Zhang, 2018), it is inevitable that a small percentage of the Earth's population will develop a detrimental mutation due to an alteration in the amino acid sequence of a protein. These gene variants have devastating effects in the form of genetic disorders; it is estimated that 3.3 million children under the age of 5 die every year from defects caused by a genetic disorder (Zarocostas, 2006).

The study of proteins is hence critical towards the advancement of human health. We need to identify the source of genetic disorders; that is, the localisation of genetic constraints on protein structures. The field of bioinformatics has led to an increased understanding of proteins and their functions, as bioinformatics is concerned with the collection, storage and analysis of complex biological data and information using

computer technology (National Human Genome Research Institute, 2022). However, this report aims to specifically focus on the quantification and visualisation of genetic constraints in protein regions through the application of machine learning techniques. Machine learning is a discipline driven by artificial intelligence which uses data and algorithms to imitate and gradually improve on the way that humans learn (IBM Cloud Education, 2020). Insights derived from machine learning are entirely data-driven, which is especially relevant to this project as the quantification of genetic constraints is an entirely data-driven approach.

The Garvan Institute is patient-focused and their researchers are dedicated to their mission in using cutting-edge technology and collaborating with researchers to harness all information encoded in the human genome to diagnose, predict and prevent disease (Garvan Institute of Medical Research, 2021). The pathogenicity of protein alterations are associated with functional importance of protein regions, making it imperative to analyse the spatial distribution of proteins. The recent explosion of human population-based sequencing studies in conjunction with the number of solved structures deposited in the Protein Data Bank (PDB) facilitates a data-driven approach to uncover the principles of protein operation (Sivley et al. 2018). These structures in the PDB are visualised using Uniprot and Icn3d which portray an accurate image of each protein and their molecular structure.

A previously published and validated model by Karczewski et al. (2020) provides us with the expected number of mutations for each amino acid of a protein. A model is then constructed to quantitatively estimate a proportion of the expected value against the observed number of mutations which is obtained from a larger genomic database of more than 140,000 humans. This study compiles data from the Genome Aggregation Database (gnomAD) and produces high-confidence Predicted Loss-of-Function (pLOF) variants to predict tolerance in inactivation of protein-coding genes. Although the scope of Karczewski's research paper involves considering the phenotypic consequences of functionally disruptive mutations for protein-coding genes, this approach provides us with a decisive approach on quantifying protein constraints. The similarity between our methodology and this research paper revolves around scrutinising the dataset at a molecular level and understanding the link between proteins, their structure and how mutations are formed.

Another research paper which has aided our analysis looks at compositional data analysis of proteins by O'Brien et al. (2018). Proteome quantitation is important to characterise protein expression across various datasets and mass spectrometry is an accessible tool for it. However, this research paper focuses on isobaric labelling, which enables precise quantification of protein expression. Statistical models of compositional proteomics were evaluated in terms of accuracy, sensitivity and specificity. Models of constrained and unconstrained protein structures were compared based on spatial composition which is quite similar to our approach. The evaluation metrics proposed in this paper are useful for unsupervised learning methods in machine learning. These algorithms uncover hidden patterns within the data devoid human intervention. Its ability to discover similarities and variation in information makes it an ideal method to cluster proteins based on its structural composition(IBM Cloud Education, 2020b). Although this research is largely quantitative with various statistical and mathematical formulas used, our project primarily consists of a machine learning approach on proteins. Overall, the dichotomy between the aforementioned research papers and the methodologies used in this report is that we've primarily used clustering methods to achieve the objective of grouping parts of proteins together based on proximity and identifying the constrained zones.

3. Scope

This report will undergo a multi-stage process:

- Stage zero: Exploratory data analysis is first conducted to gain an understanding of amino acid location and spatial distribution.
- Stage one: Amino acids will be grouped based on proximity, which will be performed through various clustering techniques; namely, K-Means clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Agglomerative Hierarchical Clustering (AHC)
- Stage two: Each method will then be evaluated through their ability to minimise the average gamma values across different clusters of proteins, as minimisation of the gamma value results in the maximisation of the potential areas where we can detect genetic constraint. This second-stage evaluation will be performed through use of the Silhouette score, the Calinski-Harabasz (CH) Index, and the Davies-Bouldin Index (DBI).

Deliverables for this project include a viable model which quantifies areas of genetic constraint within proteins based on the information contained in the datasets, along with this report explaining the methods used to construct the model. Protein regions with genetic constraints will be visualised based on gradient scales using multiple algorithms in a 3D diagrammatic representation. A holistic evaluation of the different algorithms and methods used will then be conducted to identify better and more appropriate models. To accompany our report, an audio-visual presentation (12 minutes) will be included to further guide relevant stakeholders through our methodology, findings, and potential business impact.

The datasets used for this project will be based on client-provided datasets relating to amino acids in proteins. Each dataset corresponds to a particular protein which includes the spatial coordinates and gamma value of each amino acid, along with the observed and expected number of mutations. The following assumptions are observed with regards to the dataset:

- That the gamma output in the dataset is to be used as validation data and not directly as independent variables in order for this model to be viable for forecasting purposes.
- That if two amino acids are in close contact, mutations in one are likely to be followed by mutations of the other, as preserved by its 3D structure.

We have used the O43526 protein for the majority of our analysis (unless stated otherwise) so that results are kept consistent with minimal chance for error.

4. Methodology

Considering the objectives mentioned in the scope, we believe clustering methods are ideal in grouping the amino acids and determining statistical significance of each model whilst comparing these methods through other evaluation metrics. Beginning with preprocessing, we've initiated the preliminary stages of understanding the data first.

4.1. Preprocessing

Data preprocessing involves improving the usability and readability of the data; there are 4 key procedures for data to be successfully preprocessed (Mesevage, 2021) and ready for analysis explained below.

4.1.1. Data quality assessment

Upon assessing the quality of the data, there are no missing data fields so data cleaning is not required. Data outliers can have a high impact on data analytics results, however the scope of our project does not necessitate that outliers are problematic. Finally, the 10 protein datasets are in .tsv (tab-separated value) format, however Python's pandas library is able to convert into .csv (comma-separated value) without any problems.

4.1.2. Data transformation

In terms of transforming the data, the only procedure that aligns with our scope in delivering implementations of unsupervised learning (clustering methods) is feature selection. This is the process where a decisive approach is taken to account for certain variables that will be useful for certain parts of our analysis. For example, when calculating the distance between each amino acid, we may select only the cartesian coordinates (*x, y, z*) of the entire dataset. Ultimately, having unnecessary features that do not assist in some of our methods may result in lower accuracy and may increase the training process.

4.1.3. Data reduction

Data reduction for our project involves the involvement of only the O43526 protein. The only exception is the data visualisation of the protein structures to showcase the dichotomy between the arrangements of amino acids. We will be able to fulfil the client's requirements using one protein lattice. It is possible that the quality of analysis could be improved through the use of all proteins to broaden the project's insights; however, accounting for simplicity and readability means that we will adhere to only one protein.

4.2. K-Means Clustering

K-Means clustering is an iterative, unsupervised learning algorithm which divides an unlabeled dataset into different clusters, where *K* defines the number of predefined clusters. For an effective clustering method, we assign each data point to the nearest *k* centroids, then update each centroid to reflect any points that have been re-assigned. A centroid is defined as the mean or weighted average of the points in a cluster.

P(i) is the cluster assigned to element *i*, *c(j)* is the centroid of cluster *j*, $d(v_1, v_2)$ the Euclidean distance between feature vectors v_1 and v_2. The goal is to find a partition P for which the error (distance) function is minimum, which is at

$$E_p = \sum_{i=1}^{n} d(i, c(P(i)))$$

.

Step-by-step algorithm:

Assume that we are given a feature-vector matrix $X^{n \times p}$:

1. Start with an arbitrary partition *P* of *X* into *k* clusters
2. For each element *i* and cluster $j \neq P(i)$ let E_p^{ij} be the cost of a solution in which i is moved to j:

   if $E_{p^*}^{i^*j^*} = min_{ij} E_p^{ij} < E_p$ then move i^* to cluster j^* and repeat step 2 else halt. (Koren and Bell, 2010)

Figure 1: Demonstration of K-Means Clustering

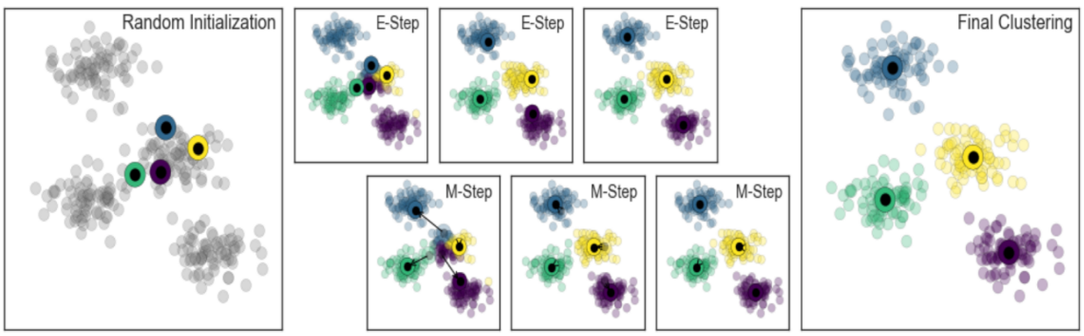


Figure 1 (VanderPlas, 2019) displays a visualisation of K-Means clustering which will be performed on our proteins dataset. The issue now arises from selecting the appropriate number of clusters, *k*. Fortunately, there are various metrics that can be utilised, including the silhouette score, Calinski-Harabasz Index and Davies-Bouldin Index. Importantly, these metrics assess clustering methods, but do not provide a predetermined value of *k*. Let's now explore the theoretical aspect of these metrics to clarify the need for an appropriate *k* value.

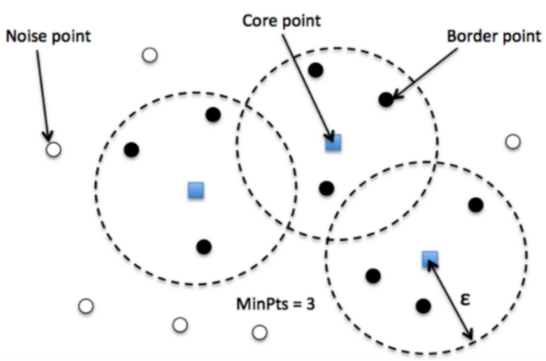
4.3. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

In the context of unsupervised learning, a density-based clustering algorithm refers to multiple methods that identify distinct groups of data based on contiguous blocks of high point density, separated from neighbouring blocks of low point density.

Density Based Spatial Clustering of Applications with Noise is referred to as DBSCAN. In this algorithm, densely populated data points are grouped into a single cluster based on their density. In a large spatial dataset, which contains noise and outliers, it can identify clusters of arbitrary shapes and varying densities. Moreover, DBSCAN allows noise to be detected in the model unlike other clustering algorithms, and it is robust to outliers.

DBSCAN only requires two parameters: epsilon and minPoints. In this algorithm, the epsilon represents the circle whose radius equals the size of the epsilon. This circle is used to categorise data points into the core, border, or noise categories. A data point is considered core if the circle around the point also contains at least 'minPoints' number of points. If the number of points in a circle is less than minPoints, then the data point is classified as a Border point, however, if there are no other data points besides the data point within epsilon radius, then it is considered as Noise (Chauhan, 2022).

Figure 2: Demonstration of DBSCAN



DBSCAN locates the data points in space using the Euclidean distance method (such as using KNN model). The value of the minPoints should be greater than the number of dimensions of the dataset;

$$minPoints \geq Dimensions + 1 .$$

In addition, the epsilon value is calculated using K-distance graphs. The maximum curvature, also known as the elbow point, of the k-distance graphs is considered the value of epsilon. The K value in the K distance graph is found by

$$k = minPoint - 1 .$$

Step-by-step algorithm

1. The algorithm arbitrarily chooses a point on the dataset.
2. The algorithm draws a circle with epsilon radius around the data point. If there are at least 'minPoint' points within the circle then all these points are part of the same cluster.
3. The clusters are then expanded recursively, repeating steps 1 and 2 on the neighbouring points.

4.4. Agglomerative Hierarchical Clustering (AHC)

Hierarchical clustering algorithms are either agglomerative i.e. bottom-up, or divisive, i.e. top-down (JavaTpoint, 2021). In practice, hierarchical agglomerative methods are often used - efficient exact algorithms are available, but more importantly to users, the dendrogram, or tree, can be visualised. The step-by-step algorithm for AHC is described below.

Step-by-step algorithm

1. Given each data point, we treat them as a single cluster. For example, if we have N data points, then there are also N clusters.
2. We group two data points that are closest to each other and merge them into a cluster, thus giving us N - 1 clusters.
3. The previous step is repeated with the two closest clusters and they are then merged, leaving us N - 2 clusters.
4. Step 3 is then repeated until we achieve one giant cluster of all the data points.

To visualise AHC we typically use a dendrogram and in particular we will be interested in analysing the distance between each cluster of chained amino acids for each protein. In addition, the Euclidean distance metric will be used to form these clusters similar to aforementioned clustering methods.

4.5. Evaluation metrics for clustering methods

Based on our objective to locate areas of genetic constraint in the protein, our clustering methods commonly group amino acids in a manner where we aim to minimise the average gamma values (obs/exp) across different clusters. This strategy is adopted so as to maximise the areas of the protein regions where genetic constraint can be detected. We also closely monitor the variance of gamma values across each cluster to maintain groupings of amino acids with similar gamma output.

Whilst critically analysing the gamma values of each cluster, we can evaluate the performance of the clustering methods as shown below.

4.5.1. Silhouette score

The Silhouette score is a metric to evaluate the performance of clustering methods. Silhouette analysis requires us to analyse the separation distance between the clusters and these distance metrics can be any one of the following: Euclidean, Manhattan, Minkowski or Hamming. In our report, we obtain a value known as the Silhouette Coefficient which ranges between -1 and 1 and is calculated using the mean of the distance between intra and inter clusters. Mathematically, this metric can be represented by the following equation (Joshi, 2021):

$$s_i = \frac{b_i - a_i}{max\left(b_i, a_i\right)},$$

where, b_i is the inter-cluster distance which is the average distance between the closest cluster of datapoint *i*:

$$b_i = \min_{k \neq i} \frac{1}{\left|C_k\right|} \sum_{j \in C_k} d(i,j),$$

whereas, a_i is the intra-cluster distance which is the average distance to all other points:

$$a_i = \frac{1}{\left|C_i\right| - 1} \sum_{j \in C, i \neq j} d(i,j).$$

If the Silhouette coefficient is 0 then it indicates that the data is on or very close to the decision boundary between two neighbouring clusters. On the other hand, a negative score indicates that the data has possibly been assigned to the wrong cluster (Mehta, 2022).

4.5.2. Calinski-Harabasz (CH) Index

The Calinski-Harabasz (CH) Index is another evaluation metric for clustering methods which is calculated as a ratio of the sum of inter-cluster dispersion and the sum of intra-cluster dispersion for all clusters. This metric is based on the principle of variance ratio. The following steps illustrate the calculation of the Calinski-Harabasz Index (Sidyakov, 2022).

Step 1: Inter-cluster dispersion calculation

The inter-cluster dispersion between clusters is known as the group sum of squares (BGSS). The BGSS measures the weighted sum of squared distances between the centroids of a cluster and the centroid of the entire dataset, given by the following formula:

$$BGSS = \sum_{k=1}^{K} n_k \times \| C_k - C \|^2,$$

where n_k is the number of observations in cluster *k*, *C* is the centroid of the dataset and C_k is the centroid of cluster *k*.

Step 2: Intra-cluster dispersion calculation

Next, we calculate the intra-cluster dispersion, also known as the within group sum of squares (WGSS). The WGSS measures the sum of squared distances between each data point and the centroid of the same cluster. We compute the WGSS at k, given by the formula:

$$WGSS_k = \sum_{i=1}^{n_k} \| X_{ik} - C_k \|^2,$$

where n_k is the number of observations in cluster k, X_{ik} is the i-th observation of cluster k and the centroid of cluster k given by C_k. The calculation of WGSS at k is calculated by the formula

$$WGSS = \sum_{k=1}^{K} WGSS_k.$$

Step 3: Calinski-Harabasz index calculation

Finally, the CH index is calculated as the ratio of the sum of inter-cluster dispersion over the sum of intra-cluster dispersion for all clusters. The calculation thus becomes

$$CH = \frac{\frac{BGSS}{K-1}}{\frac{WGSS}{N-K}} = \frac{BGSS}{WGSS} \times \frac{N-K}{K-1},$$

where N is the total number of observations and k is the total number of clusters.

The CH index, also known as the Variance Ratio Criterion, performs optimally when the score is higher.

4.5.3. Davies-Bouldin Index (DBI)

The Davies-Bouldin Index (DBI) evaluates the goodness of split by a K-Means clustering algorithm for various given clusters. The following contains a step-by-step calculation of the DBI metric (PyShark, 2021).

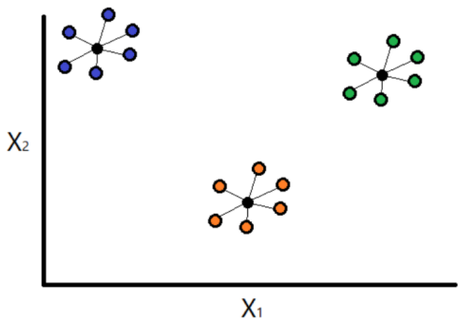
Step 1: DBI intra-cluster dispersion calculation

Similar to the CH index aforementioned, we first calculate the dispersion of cluster at i:

$$S_i = \left\{ \frac{1}{T_i} \sum_{j=1}^{T_i} \left| X_j - A_j \right|^q \right\}^{\frac{1}{q}},$$

where i is an identified cluster, T_i is the number of observations in cluster i, X_j is the j-th observation in cluster i and A_i is the centroid of cluster i. To calculate this value, we take the average distance between each observation within a cluster and its centroid. The plot below should illustrate this concept, however it is important to note that the assumption is that we are using K-Means clustering as our chosen method.

Figure 3: K-Means clustering with 3 centroids (indicated by the black centred dots)



Step 2: Separation measure calculation

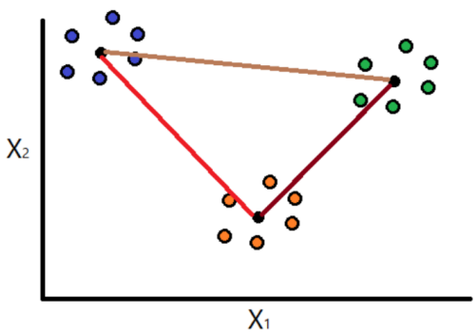
The separation calculation between clusters i and j, defined by Davies, D., & Bouldin, D. (1979) is given by:

$$M_{ij} = \left\{ \sum_{k=1}^{N} \left| a_{ki} - a_{kj} \right|^{p} \right\}^{\frac{1}{P}},$$

$$M_{ij} = \left\{ \sum_{k=1}^{N} \left| a_{ki} - a_{kj} \right|^{p} \right\}^{\frac{1}{P}} = \left\| A_i - A_j \right\|_p,$$

where a_{ki} is the *k*-th component of the centroid A_i, a_{kj} is the k-th component of the centroid A_j and N is defined as the total number of clusters. To visualise this calculation, we effectively calculate the distance between each centroid and if $p = 2$ for the above formula, then we are calculating the Euclidean distance between clusters *i* and *j*.

Figure 4: Visualisation of Euclidean distance when using K-Means Clustering



Step 3: Similarity between clusters calculation

Now we calculate the similarity between clusters *i* and *j*:

$$R_{ij} = \frac{S_i + S_j}{M_{ij}},$$

where S_i is denoted by the intra-cluster dispersion of cluster *i*, S_j is the intra-clustered dispersion of cluster *j* and M_{ij} is the distance between the centroids of clusters *i* and *j*. Computing the similarity between these clusters is calculated through the sum of two intra-cluster dispersions divided by the separation measure.

The greater the value for R_{ij}, the more similar the clusters i and j are. Finally, we calculate the highest ratio from all R_{ij} where $i \neq j$ and thus we take the average of the similarity measures for each cluster to get

$$\overline{R} = \frac{1}{N} \sum_{i=1}^{N} R_i .$$

As a result of these calculations, we can thus interpret the DBI to perform optimally when the score is lower.

5. Findings

All implementations of code can be navigated in the GitHub repository located at Appendix 8.1.

5.1. Exploratory Data Analysis (EDA)

Prior to our analysis, conducting EDA is essential for various reasons including understanding the key understandings of the dataset, providing insights of the descriptive statistics of the data, exploring the data types and structure of the dataset and most importantly, for our approach, producing 3D plots for each protein lattice. Exploring the data is imperative for our approach, however, visualisation of the Cartesian coordinates in these proteins can yield some interesting discoveries, including clustered and non-clustered areas of amino acids. Interpreting each amino acid can be quite difficult in a 3D plot, hence it is justified to examine a 2D version of each protein to analyse the spatial distribution of the amino acids more clearly. A key assumption stated in the scope which is maintained through this project is that we've only used the O43526 protein in our analysis, unless stated otherwise. This is to just keep the initial data exploration simple; additionally, the other 9 proteins yield similar results. Importing the dataset into Python for analysis, we've observed that the data contains no null or missing values and in general the data has consistent scaling and there are no inconsistencies for each feature, thus indicating that data standardisation is not a requirement.

Figure 5: Distribution of x coordinate (O43526)

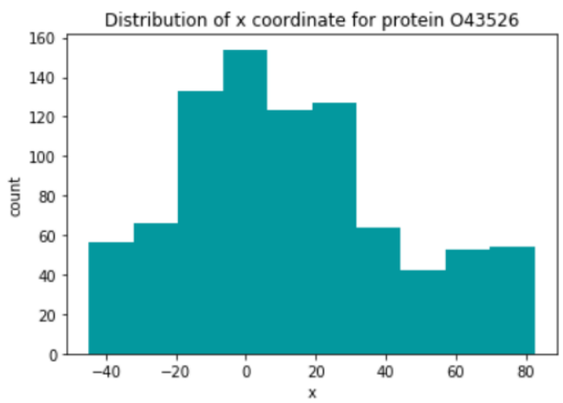


Figure 6: Distribution of y-coordinate (O43526)

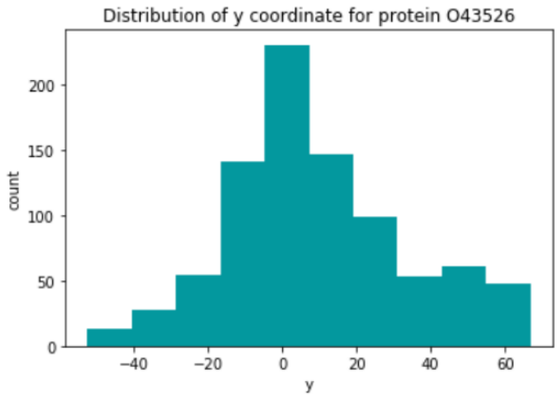
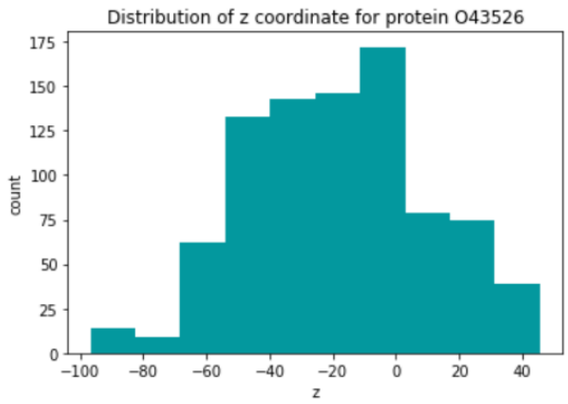


Figure 7: Distribution of z coordinate (O43526)



To start the EDA, we've implemented some simple histograms of the Cartesian coordinates of the protein structures. Though this does not constitute any meaningful insights, we can observe an initial understanding of where most of the amino acids are located at. For example, Figure 5 has a large amount of amino acids spread across horizontally, whereas for the y and z coordinates there are fewer amino acids. The distribution for all figures is multimodal.

For the same protein, we are able to extract the descriptive statistics as shown in Appendix 8.2. Despite there being few noteworthy results on this table, we observe the mean expected number of mutations for each amino acid to be 0.56, whereas the proportional decrease as an estimate of genetic constraint is given by the gamma value to have a mean of 0.52. We have integrated this proportion with clustering methods to indicate areas of protein constraints depending on the gamma value. Prior to plotting these proteins, we finish the preprocessing section with an initial inspection of the key features in the dataset, by comparing the correlation coefficients in Appendix 8.3. One misleading aspect of this correlation matrix is the inclusion of the index variable, since it's just a counter for the number of rows in the dataset, hence we ignore these values. In summary, there is no correlation between any of these variables as the value of R does not approach near 0.7, indicating there are no multicollinearity issues.

We divert our analysis to the expected value of mutations, denoted in the dataset as 'exp' and the gamma values, which are the true unobserved proportional depletion in mutation. This is imperative for our initial understanding of genetic constraints within different regions of proteins.

Figure 8: Expected mutations on Protein O43526

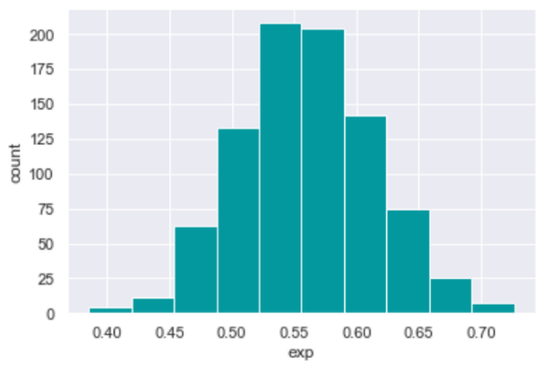


Figure 9: Expected mutations on Protein P01009

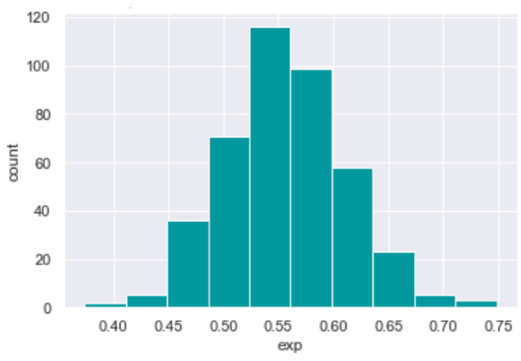


Figure 10: Expected mutations on Protein P06576

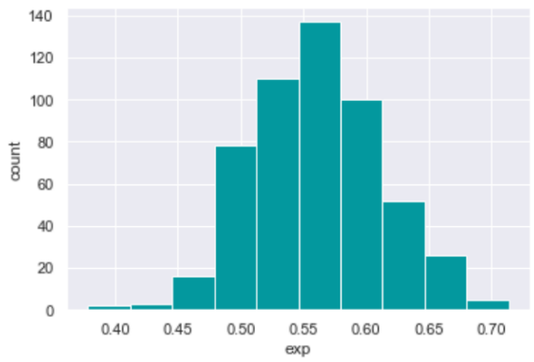
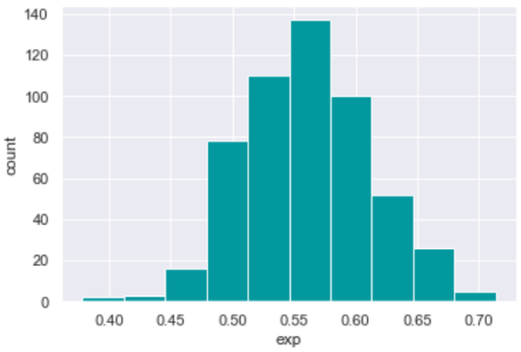
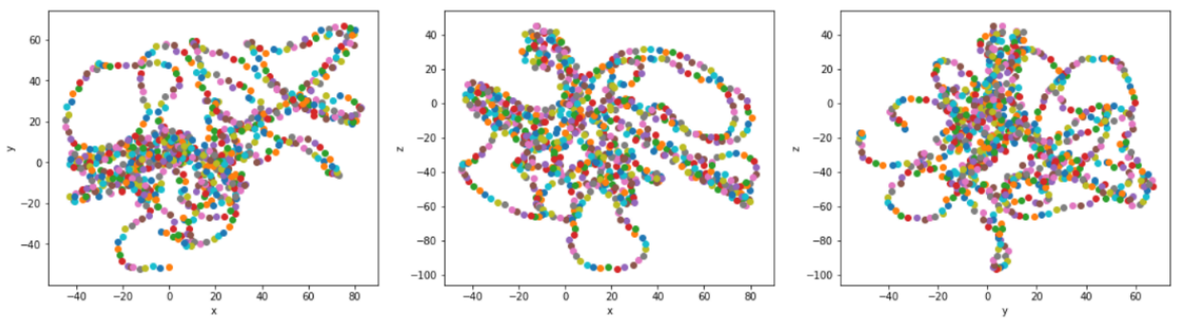


Figure 11: Expected mutations on Protein P17181



Upon inspection, Figures 8-11 consists of the expected value for genetic mutation across 4 different proteins. The histograms indicate a normal distribution across all of these plots, thus indicating that data normalisation is not required as all of the values fall between 0 and 1. A common feature between these histograms is that the mean expected value is 0.56, meaning on average there are 0.56 mutations for each amino-acid. Performing a basic foundation of our understanding of the data, we can transform the cartesian coordinates into plots that will be useful for interpretation, beginning with the spatial distribution of these proteins.

Figure 12: Spatial Distribution of Protein O43526
From left to right: plot of x and y coordinates, plot of x and z coordinates, plot of y and z coordinates



Here in Figure 12, we observe the spatial distribution of the O43526 protein comprising all the amino acids and we are able to understand the areas of clustered and non-clustered molecules. This is pivotal for our clustering methods, because we can identify areas that are clustered and determine proximity between amino acids. Despite the usefulness of observing the protein structure in a 2D view, a 3D view will constitute more insights for clustering and classification.

A 3D plot of a few chosen proteins are also provided below for reference:

Figure 13: 3D plot of Protein O43526

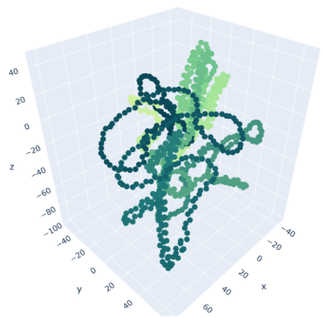


Figure 14: 3D plot of Protein P06576

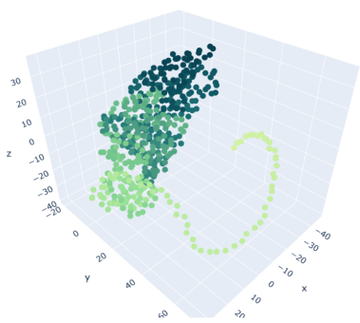


Figure 15: 3D plot of Protein P69905

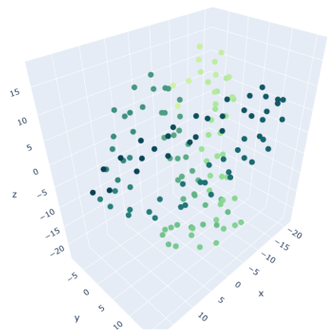
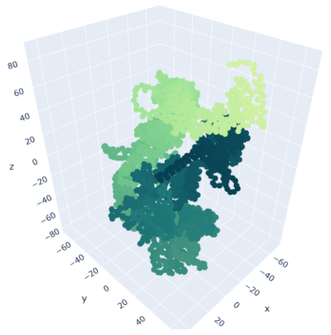


Figure 16: 3D plot of Protein Q5S007



The remaining proteins are provided in Appendix 8.4. Upon inspection, we observe various different structures of 4 selected proteins. Figure 16 displays a protein which appears to have amino acids in close proximity, indicating that the distance between each amino acid is much smaller compared to the other plots. Figure 15 then shows a large spatial distribution of the protein with most of the amino acids scattered across large distances. Figure 13 contains a combination of compressed amino acids and scattered and Figure 14 has a distinct shape, with a tail-like structure which expands the protein across. To confirm the validity of these plots, we compared them to the plots produced in iCn-3D (Appendix 8.11).

5.2. K-Means Clustering

The K-means clustering algorithm calculates the gamma values using the cartesian coordinate features of the dataset. These features are used to construct clusters in the model, and the gamma values are then calculated using the clusters. For each cluster, the average expected value of mutation and observed value of mutation are calculated, which is then used to calculate the gamma value of the cluster. The gamma value of the model is determined by taking the mean of the gamma values of each cluster. The number of clusters to use is determined by comparing the gamma value of our model with the gamma value of other models. For a model, the optimal K value is the one that produces the lowest gamma value. In this algorithm, we have kept the number of clusters to a minimum, between 2 and 5 that will produce the minimum gamma value chosen by the optimal number of clusters k.

After implementing the K-Means Clustering method on the proteins dataset, Figure 17 below is produced with a total of 5 clusters. Initial inspection reveals that the blue and brown regions in the figure are densely populated compared to the rest, indicating that certain amino acids are close to one another. Due to the close proximity, they have a greater number of neighbouring points compared to other amino acids. Moreover, the blue and brown clusters have lower gamma values than the yellow cluster with the fewest data points. As an example, the brown region has a gamma value of 0.5484 while the blue region has a gamma value of 0.5791, which is lower than the yellow region with a gamma value of 0.6403. Despite being densely populated, the blue and brown clusters indicate that their regions are more constrained than the yellow region. Additionally, the green region, although densely populated, has the lowest gamma value of 0.3636, compared to all other clusters. Thus, the green region represents a significant genetic constraint, that is, mutations are most likely to be depleted in the green region.

Figure 17: K-Means Clustering on Protein O43526

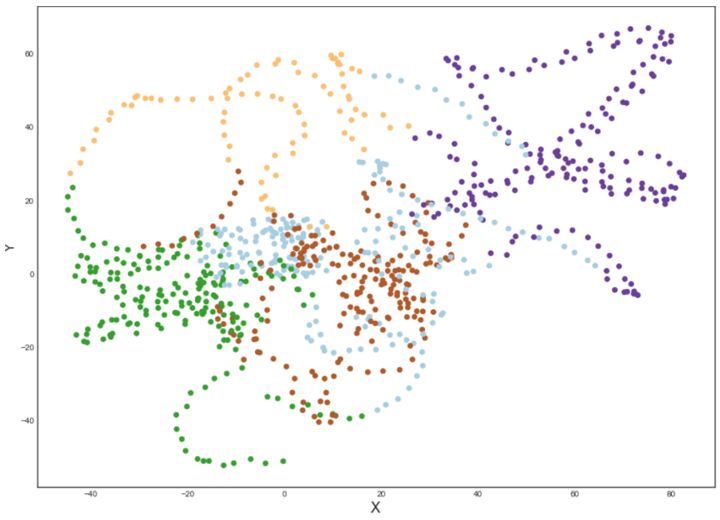


Table 1: Gamma output of clusters 1 to 5 using K-Means Clustering

Total minimum average gamma value	0.5088514887119936
Number of clusters	5
Cluster 1 (Blue)	0.5791013016495951
Cluster 2 (Brown)	0.5483514848437369
Cluster 3 (Green)	**0.36359038544022826**
Cluster 4 (Purple)	0.4128672537324598
Cluster 5 (Yellow)	0.6403470178939478

5.3. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

The DBSCAN groups protein structures based on their *x*, *y*, and *z* coordinates. By calculating the gamma value for various minPoints, the algorithm is able to determine which minPoint minimises the gamma value of the protein structure. Additionally, the model calculates the distance between neighbouring points in the protein structure based on a K-distance graph. Based on the distance matrix obtained by the K-distance algorithm, the optimal epsilon value is then determined for the corresponding minPoint value. Using the Knee and Elbow method, epsilon is calculated by identifying the maximum curvature point on the Knee point graph. The Knee point graph in Figure 18 below illustrates that the maximum curvature point, or epsilon value, is at 10.6937. Thus, the DBSCAN produces clusters of varying sizes when applied to this dataset. Consequently, the optimal minPoint value that minimises the gamma value of the model is selected by comparing the gamma value produced with different minPoint values.

Figure 18: Knee Point curve

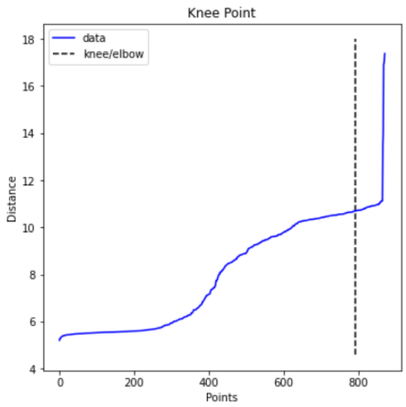
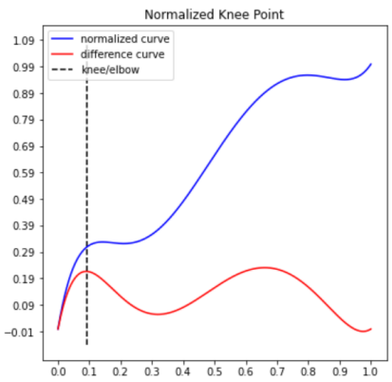


Figure 19: Normalised Knee Point curve



Cluster scatter plots demonstrate that the green and blue clusters are densely populated across all points. The reason for this can be attributed to a large optimal epsilon value being chosen in the model. With increasing epsilon values, the distribution of data points becomes more homogeneous, that is, there are fewer clusters grouped with more data points being grouped together. In reference to Figure 20, it is noteworthy that despite the yellow region containing a small cluster, it produces the highest gamma value in comparison with other clusters, indicating that the region contains low levels of mutation depletion. Similarly, blue and green clusters also produce relatively high gamma values. Therefore, the purple cluster with a very low gamma value is the high constraint zone of the model.

Figure 20: DBSCAN on O43526 Protein

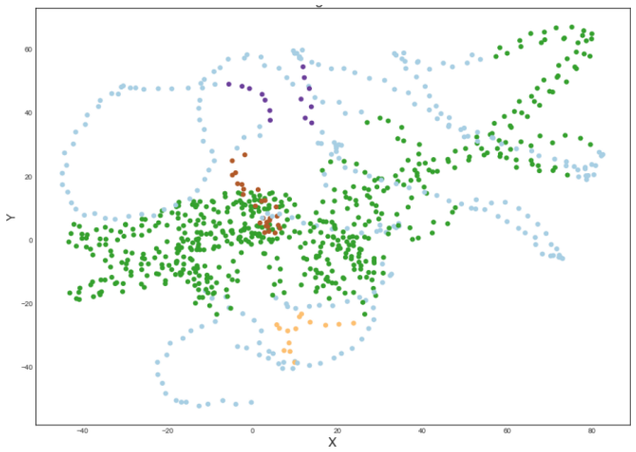


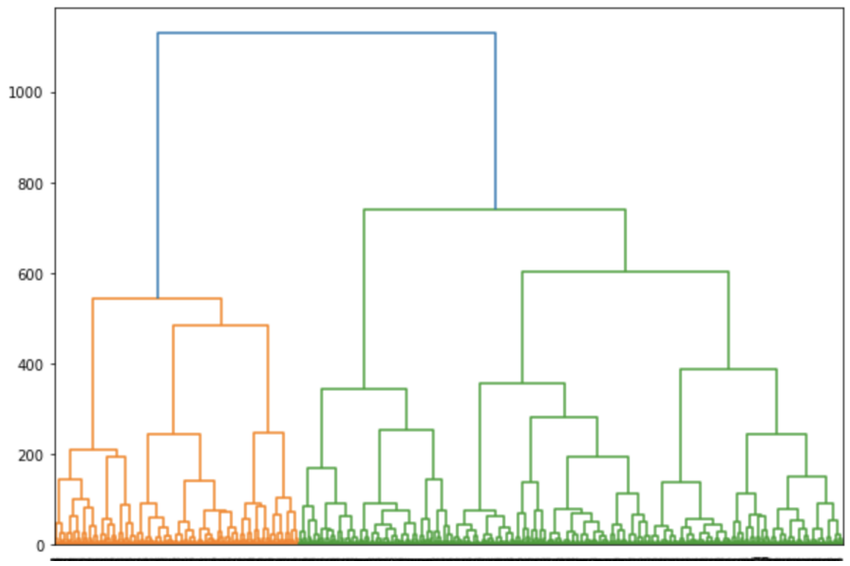
Table 2: Gamma output of clusters 1 to 5 using DBSCAN

Total minimum average gamma value	0.4288679978074875
Number of clusters	5
Cluster 1 (Yellow)	0.5963903396776727
Cluster 2 (Blue)	0.5057785914437982
Cluster 3 (Brown)	0.3826722554873264
Cluster 4 (Green)	0.5259469283720811
Cluster 5 (Purple)	**0.13355187405655922**

5.4. Agglomerative Hierarchical Clustering (AHC)

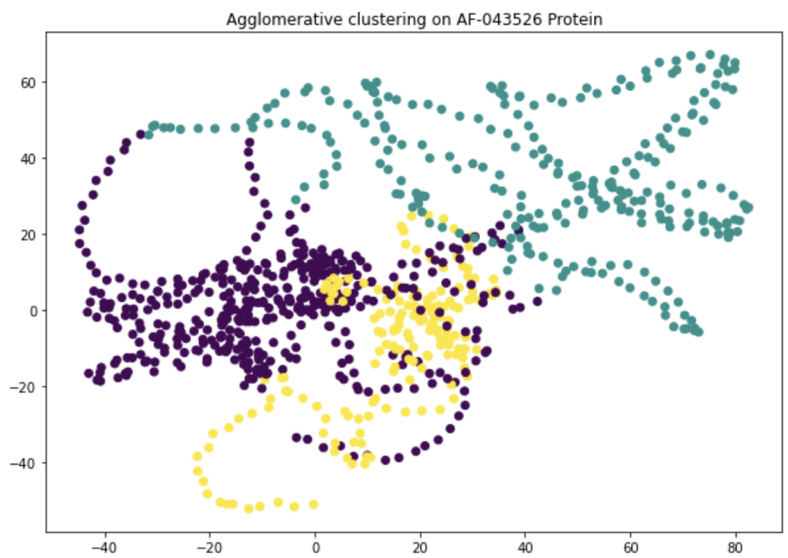
Although K-Means clustering provides us with an effective implementation of grouping the amino acids to determine genetic constraints, the AHC method uses a bottom-up approach where each data point starts in its own cluster. This is a greedy algorithm that will sequentially select data points, in our case the amino acids. Beginning with the data frame of the Cartesian coordinates of each amino acid, we utilise the Euclidean distance metric to measure the closeness of each amino acid in terms of both its spatial distance and gamma value (expected/observed). To grasp an initial understanding of the clusters in a hierarchical view, a dendrogram becomes useful.

Figure 21: Dendrogram of AHC on Protein O43526



To interpret this dendrogram of the O43526 protein we focus on the height at which the amino acids are joined together. Observations (amino acids) near the base are grouped closely together since the height of the link that joins them together is the smallest (Bock, 2018). As a result, cutting the dendrogram at its longest length, will allow us to more accurately determine the number of clusters required. Based on this, clustering can be most appropriately done using 3 clusters. Performing AHC using different linkage methods with the chosen number of clusters, we then obtain the below 2D version of the protein.

Figure 22: Agglomerative Hierarchical Clustering on Protein O43526



The clear distinction between the density of the yellow cluster and the rest of the clusters hints to us that there are various parts of the protein where the amino acids are tightly bound together. The sub-clusters in comparison are held even closer to each other with low Euclidean distance. Conversely, the green cluster is wider, as shown in this dendrogram, indicating amino acids molecules that are far away from each other. To confirm this observation, we look from the perspective of a scatter plot.

Visualising the protein structure with the incorporated clusters, we observe that the yellow region is tightly clustered with amino acids having much lower euclidean distance from each other, whereas the green and purple regions are more dispersed. We can hypothesise that this yellow region is more likely to be depleted in mutation compared to the other clusters. This is evidenced by the table below which depicts that the yellow cluster has a minimised gamma value in comparison to the other clusters, representing a region of genetic constraint.

The method has been attempted with a varying number of clusters and different distance linkage methods. Using the 'Ward linkage method' with 3 clusters has allowed us to yield the total minimum average gamma value.

Table 3: Gamma output of clusters 1 to 3 using AHC

Total minimum average gamma value	0.493880
Number of clusters	3
Cluster 1 (Purple)	0.529787
Cluster 2 (Green)	0.605980
Cluster 3 (Yellow)	**0.345873**

5.5. Evaluation

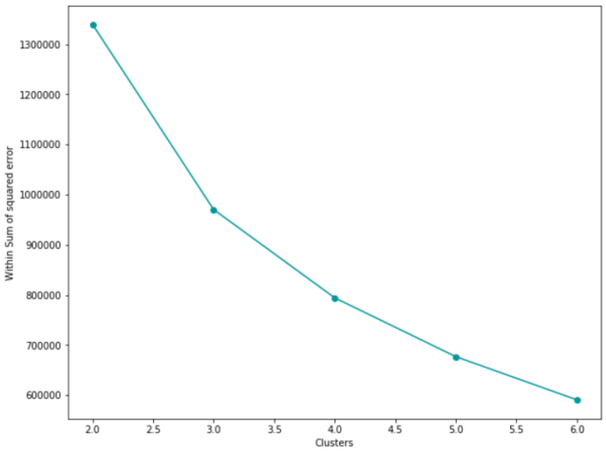
5.5.1. Silhouette Score Analysis

Clustering is an important machine learning technique that plays a vital role in identifying genetic constraints of proteins (Hayasaka, 2022). Unfortunately, in many instances the number of clusters we require for analysis is quite ambiguous unless there is domain knowledge of the dataset that directly assists in the determination of the number of clusters. Utilising a brute-force approach in determining the correct number of clusters can be an effective method, however there are various evaluation metrics that can not only detect the appropriate number of clusters, but also assess cluster quality. Moreover, these evaluation metrics are paramount in comparing the performance and of each clustering method we've used. It has to be established that Section 5.5.1 contains interpretation of evaluation metrics only for K-Means clustering, Section 5.5.2 for DBSCAN and Section 5.5.3 for AHC, however the full tables of comparison can be viewed in Section 5.5.4.

We begin our evaluation with the introduction of the silhouette score, where the other metrics have been calculated in Appendix 8.5. We've selected clusters of $k = 2, \ldots, 6$ and we witness the Silhouette coefficient to be the lowest when $k = 4$ resulting in a Silhouette score of 0.308 and the highest for $k = 5$, approximately 0.331. In the context of our project, we need to identify the localisation of the amino acids that are clustered together, hence a silhouette score closer to 1 is more desirable; in this case the silhouette coefficient

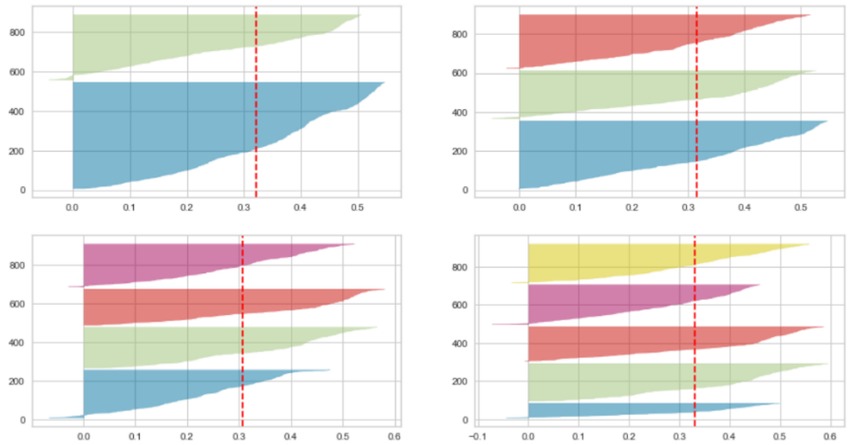
achieved at $k = 5$ is appropriate. An inspection of the within sum of squared error at each cluster is ideal because this error measures the difference between the predicted value and the sample mean of the clusters.

Figure 23: Elbow curve of within sum of squared error versus clusters



Here, Figure 23 illustrates an elbow plot of the within sum of squared errors across the range of clusters we've selected. Heuristically, we can determine the clusters required for Protein O43526 by inspecting the most dramatic decrease in the within sum of squared error, therefore, ideally we should be selecting 3 clusters. This supports our analysis of the average minimum gamma value in Table 1, where cluster 3 produces the lowest value, reinforcing our results that genetic constraint is most likely to be depleted in this region. A continuation of the silhouette analysis of Protein O43526 is shown below where using the yellowbrick.cluster library in Python and importing the Silhouette Visualizer has given us some insightful information regarding choosing the number of clusters for K-Means.

Figure 24: Silhouette plots for K-Means clustering (at k = 2,..., 5)



The plot in Figure 24 above showcases the silhouette plots as we perform a comparative analysis to determine the best value of *K*. The x-axis represents the silhouette scores with the dotted red line representing the average silhouette score for *k* clusters. Each region represents the number of clusters; for example, the top left plot has 2 clusters and the top right contains 3 clusters. One particular aspect we can highlight from these plots is that all the clusters are above the threshold (average silhouette score). Incongruous to this, we look at the fluctuations of each region and confirm that the plot for $k = 1$ has a wide and slightly inconsistent thickness of the silhouette plot. Now comparing the plots for $k = 3$ and $k = 4$, the blue cluster at $k = 3$ indicates a slight inconsistent thickness in the silhouette score, whereas the latter

plot displays relatively consistent scores. Inspecting the plot for $k = 5$, we see all the silhouette scores still pass the threshold comfortably, alluding that the optimal k-value based on the silhouette analysis is at $k = 5$. However, when comparing the analysis of obtaining the optimum k-value through the elbow method and the calculation of minimum average gamma value yields an optimal k value of 3. This is problematic as we need to choose the best possible value, which is why further analysis in other evaluation metrics and its comparison to silhouette analysis is required.

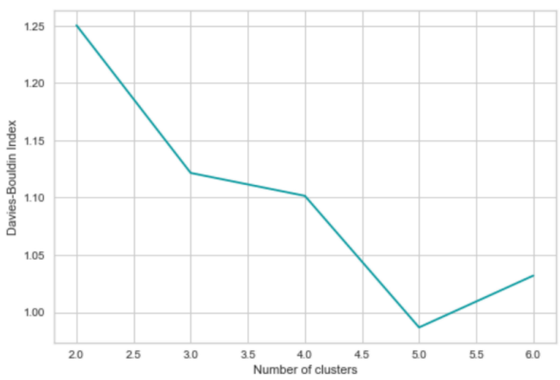
5.5.2. Calinski-Harabasz Index (CHI) Analysis

The Calinski-Harabasz Index is an evaluation metric for clustering methods which calculates the ratio between the inter and intra cluster dispersion of all the clusters. This evaluation metric signifies the ratio of the sum of intra and inter cluster dispersion and the results we have attained are indicative of densely based clusters; because of the score being relatively high, we can infer that the clusters are close in proximity. One drawback worth mentioning is that the CHI is generally higher for convex clusters than any other types of clusters, such as density based clusters mostly obtained using algorithms such as DBSCAN (Wei, 2020). To interpret this metric we expect that the higher the score, the better the clustering performance. Unlike the silhouette analysis completed in the previous section, we've computed the number of clusters k based on the minimum average gamma value which is 5. DBSCAN is a robust algorithm which is heavily determined from the parameters epsilon and MinPoints, thus we only achieve results for 5 clusters as shown in Appendix 8.6. From the results of DBSCAN in Table 2, we observe that the minimum average gamma value is 0.134 (3 d.p.). Taking 5 clusters and applying the DBSCAN algorithm results in a CHI of 11.472. In comparison to results obtained in Section 5.5.1, the Silhouette score yields the highest values for $k = 5$ clusters. We need to examine another evaluation metric called the Davies Bouldin Index to compare with aforementioned metrics and hence select the best performing clustering method for our analysis of genetic constraints.

5.5.3. Davies Bouldin Index (DBI)

The Davies Bouldin Index is a metric used to assess the similarity between clusters where the similarity is measured by comparing the distance between clusters with the size of the clusters themselves (Wei, 2020). Implementing the DBI metric on AHC and comparing these results to Silhouette score analysis and CHI will produce valuable insights in determining the feasible amount of clusters. In Appendix 8.8, we observe the evaluation metrics for AHC for $k = 2, \ldots, 6$ and we can see that the DBI output is the lowest for $k = 6$ at 1.118. This is in stark contrast with the minimum average gamma values that are achieved in Table 3, which indicates that the desired number of clusters in terms of gamma, is $k = 3$. A diagrammatic representation of the number of clusters against the DBI is shown below.

Figure 25: DBI plot for Agglomerative Hierarchical Clustering (at k = 2,..., 6)



In Figure 25, the plot highlights the value of DBI for each cluster and we immediately observe that at cluster $k = 5$, the DBI is at its lowest, however, interestingly it is at its highest highest when $k = 2$. We cannot feasibly make a decision here without taking into consideration the other metrics and its output for the 3 clustering methods we have utilised in this project. A collation of all the evaluation metrics output for all the clusters are shown below, however it is important to note that for DBSCAN, our implementation includes an epsilon value of 10.694 (3 d.p.) and *min_samples* = *7*, thus yielding the evaluation metrics results for $k = 5$ only.

5.5.4. Comparing all evaluation metrics for clustering methods

Table 4: Comparison table of metrics for K-Means Clustering (correct to 3 decimal places)

Number of Clusters	Silhouette Score	Calinski-Harabasz Index	Davies-Bouldin Index
2	0.322	458.568	1.250
3	0.316	**481.443**	1.121
4	0.308	455.841	1.101
5	**0.331**	438.215	**0.987**
6	0.324	426.495	1.032

Table 5: Comparison table of metrics for DBSCAN (correct to 3 decimal places)

Number of Clusters	Silhouette Score	Calinski-Harabasz Index	Davies-Bouldin Index
5	-0.185	11.472	4.057

Table 6: Comparison table of metrics for AHC (correct to 3 decimal places)

Number of Clusters	Silhouette Score	Calinski-Harabasz Index	Davies-Bouldin Index
2	**0.319**	**396.209**	1.288
3	0.254	351.430	1.206
4	0.243	334.041	1.202
5	0.264	336.753	1.222
6	0.282	345.717	**1.118**

To grasp an understanding of the different evaluation metrics used as well as selecting the optimal number of clusters we have constructed various tables above. Aside from DBSCAN, where our implementation produced a cluster of 5, comparisons between Tables 4 and 6 are more easily compared and will hence be examined in higher detail. The silhouette score for K-Means clustering and AHC is the highest for cluster 5 and 2, respectively, whereas the DBI is the lowest in cluster 5 for K-Means and cluster 6 for AHC. There are also differences in the CHI where for K-Means clustering the highest is for cluster 3 and for AHC the highest is for cluster 2. It is important we remind ourselves that the minimum average gamma values are the lowest for cluster 3 for K-Means, cluster 5 for DBSCAN and cluster 3 for AHC.

At the moment, there are no concrete strategies we can employ to determine the exact number of clusters, however we after assessing these metrics we can confidently state that the best clustering method to determine genetic constraint is K-Means clustering, due to the higher Silhouette scores and CHI values of 0.331 and 481.443, respectively and the lower DBI value of 0.987 compared to the other clustering techniques. The evidence shown in our results for DBSCAN, the silhouette score is -0.185, where a negative output indicates that the sample dataset has been assigned to the wrong cluster. The CHI is much lower than the other clustering methods and the DBI is slightly larger. A comparison between these evaluation metrics and the minimum average gamma values cannot be made, because here we are assessing the most favourable clustering method and the lowest gamma values yield the ideal number of clusters used. Based on this analysis, the ranking of the effectiveness of the clustering techniques we've used to determine genetic mutation are:

1. K-Means clustering
2. Agglomerative Hierarchical Clustering
3. Density-Based Spatial Clustering of Applications with Noise

6. Discussion

In our analysis of quantifying genetic constraints in protein regions, we've encountered various challenges and our approach has limitations. Since most of the results we've achieved are heavily revolved around clustering methods, a discussion of the limiting factors of these techniques is vital to address the Garvan Institute's problem statement.

According to the K-Means clustering, the minimum total average gamma value is 0.5088, which is relatively close to the middle boundary. Due to the inclusion of noise in K-Means clustering, the algorithm has difficulty grouping clusters of multiple sizes as the dataset size increases unlike DBSCAN, which is not easily influenced by noise or outliers and is also capable of handling clusters of various sizes. Based on the same number of clusters as K-Means, the minimum total average value of DBSCAN produces a gamma value of 0.4289, which is lower than K-Means and AHC. However, given our results where DBSCAN identified regions with an extremely small number of amino acids, this raises questions as to whether the model has appropriately identified the constrained zone. By covering as much of the constrained regions as possible, the model is able to be as close as possible to the true area of each constrained region (Ginni, 2022).

In AHC, certain characteristics allow for a greater flexibility of the model. In particular, the number of clusters need not be pre-specified. This is advantageous since the number of clusters of the protein is unknown. Moreover, the algorithm is easy to use and implement. However, this also follows with some drawbacks since the algorithm will not allow for an observation in a cluster to be re-assigned in further clustering iterations. Furthermore, Ward's method is also said to usually result in less than optimal clusters. In comparison to K-Means, agglomerative clustering is also about 4 times more computationally expensive. This can prove to be especially problematic, if proteins with a higher number of amino acids were to be clustered, increasing the overall time complexity of the algorithm. (San, 2016)

Based on these findings we believe that K-Means has performed the best on the basis that it has successfully identified a distinct constrained region within the protein and the chosen number of clusters has remained consistent across the evaluation metrics discussed in Section 5.5.

6.1. Recommendations

The approach adopted in our methodology attempted to find common ground between the following points:

1. Identifying models yielding regions with the lowest gamma value
2. Ensuring protein regions consist of a relatively large number of amino acids
3. Multiple algorithms pointing to similar regions of genetic constraint

From iterating through multiple algorithms, it is clear that constrained zones are locally situated within a corner of the protein.

The first recommendation would then be that genetically constrained zones can be more optimally found through multiple algorithms that succeed in identifying similar regions of mutation depletion. Appropriate algorithms to implement include a mean shift model or gaussian mixture model. A mean shift model is an unsupervised learning algorithm that is predominantly used for clustering (Yufeng, 2022). The non-parametric nature of the algorithm along with a non-necessity of predefined clusters in the feature space makes it a suitable algorithm to cluster amino acids. Similarly, gaussian mixture models can be used to classify points based on a probability distribution which can be hugely relevant in cases which involve unsupervised learning problems (Maklin, 2019). Furthermore, the inclusion of characteristics that are closely related to protein structure such as the torsion angle, can also be used as a similarity/ distance measure by future analysts approaching the problem from a structural point of view.

Our approach to the quantification of genetic constraints in protein regions using cluster analysis has significant implications on the future of healthcare. With genetic conditions and diseases such as heart disease, cancer and diabetes posing a larger threat to humanity's future than ever before, the application of machine learning techniques - such as the one we just demonstrated - will allow medical experts to harness the power of technology towards early disease diagnosis and prevention.

6.2. Response to Peer Review

Throughout the duration of this project, Protolytix has been fortunate enough to receive peer review and feedback from others while this report was in its drafting phase.

The feedback we've received from Appendix 8.9 suggests to integrate opinions and outline our experience with various models and accuracy measures. We've taken this into consideration through providing a structured and coherent implementation of the evaluation metrics for each clustering method to analyse its performance. A question is raised as to how the clustering methods we've discussed will assist in identifying the constrained zones of the proteins dataset. In response to this question, we believe the clustering methods provide us with an iterative way of viewing the distances between each amino acid where forming each cluster is an indication of potential genetic constraint. We've integrated the analysis of these clustering methods alongside the interpretation of the minimum average gamma value to quantify genetic constraints.

The feedback we've received from Appendix 8.10 expresses concern with the lack of information and description we've provided for the dataset in our report. Our team collectively agreed upon this point and we've provided a section dedicated to data pre-processing in Section 4.3. This section should alleviate the confusion on the initial assumptions we had to make prior to EDA and exploring the various clustering methods.

7. References

Bock, T. (2018). *What is a Dendrogram? How to use Dendrograms*. [online] Displayr. Available at: https://www.displayr.com/what-is-dendrogram/ [Accessed 1 Nov. 2022].

Chauhan, N.S. (2022). *DBSCAN Clustering Algorithm in Machine Learning*. [online] KDnuggets. Available at: https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html#:~:text=low%20point%20density.- [Accessed 1 Nov. 2022].

Fisher, K.M. (1985). A Misconception in Biology: Amino Acids and Translation. *Journal of Research in Science Teaching*, 22(1), pp.53–62. doi:10.1002/tea.3660220105.

Garvan Institute of Medical Research. (2019). *About the Garvan Institute | Garvan Institute of Medical Research*. [online] Available at: https://www.garvan.org.au/about-us/about-the-garvan-institute/ [Accessed 24 Oct. 2022].

Ginni (2022). *What is the difference between K-Means and DBSCAN?* [online] www.tutorialspoint.com. Available at: https://www.tutorialspoint.com/what-is-the-difference-between-k-means-and-dbscan#:~:text=K%2Dmeans%20needs%20a%20prototype [Accessed 1 Nov. 2022].

Glen, S. (2018). *Ward's Method (Minimum variance method)*. [online] Statistics How To. Available at: https://www.statisticshowto.com/wards-method/ [Accessed 1 Nov. 2022].

Hayasaka, S. (2022). *How Many Clusters?* [online] Medium. Available at: https://towardsdatascience.com/how-many-clusters-6b3f220f0ef5 [Accessed 10 Nov. 2022].

IBM Cloud Education (2020). *What is Machine Learning?* [online] www.ibm.com. Available at: https://www.ibm.com/au-en/cloud/learn/machine-learning [Accessed 21 Nov. 2022].

IBM Cloud Education (2020b). *What is Unsupervised Learning?* [online] www.ibm.com. Available at: https://www.ibm.com/cloud/learn/unsupervised-learning [Accessed 22 Nov. 2022].

Jaiswal, S. (n.d.). *Hierarchical Clustering in Machine Learning - Javatpoint*. [online] www.javatpoint.com. Available at: https://www.javatpoint.com/hierarchical-clustering-in-machine-learning [Accessed 30 Oct. 2022].

Joshi, T. (2021). *Silhouette Score*. [online] Medium. Available at: https://tushar-joshi-89.medium.com/silhouette-score-a9f7d8d78f29 [Accessed 27 Oct. 2022].

Karanam, S. (2021). *Curse of Dimensionality — A 'Curse' to Machine Learning*. [online] Medium. Available at: https://towardsdatascience.com/curse-of-dimensionality-a-curse-to-machine-learning-c122ee33bfeb#:~:text=Curse%20of%20Dimensionality%20describes%20the [Accessed 30 Oct. 2022].

Karczewski, K.J., Francioli, L.C., Tiao, G., Cummings, B.B., Alföldi, J., Wang, Q., Collins, R.L., Laricchia, K.M., Ganna, A., Birnbaum, D.P., Gauthier, L.D., Brand, H., Solomonson, M., Watts, N.A., Rhodes, D., Singer-Berk, M., England, E.M., Seaby, E.G., Kosmicki, J.A. and Walters, R.K. (2020). The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature*, 581(7809), pp.434–443. doi:10.1038/s41586-020-2308-7.

Keskin, O., Gursoy, A., Ma, B. and Nussinov, R. (2008). Principles of Protein–Protein Interactions: What are the Preferred Ways For Proteins To Interact? *Chemical Reviews*. [online] Available at: https://doi.org/10.1021/cr040409x.

Koren, Y. and Bell, R. (2010). Advances in Collaborative Filtering. In: F. Ricci, L. Rokach, B. Shapira and P.B. Kantor, eds., *Recommender Systems Handbook*. [online] Springer New York, pp.145–186. Available at: https://link.springer.com/chapter/10.1007/978-0-387-85820-3_5 [Accessed 30 Oct. 2022].

Maklin, C. (2019). *Gaussian Mixture Models Clustering Algorithm Explained*. [online] Medium. Available at: https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e [Accessed 22 Nov. 2022].

Mehta, S. (2022). *A tutorial on various clustering evaluation metrics*. [online] Analytics India Magazine. Available at: https://analyticsindiamag.com/a-tutorial-on-various-clustering-evaluation-metrics/ [Accessed 27 Oct. 2022].

Mesevage, T.G. (2021). *What Is Data Preprocessing & What Are The Steps Involved?* [online] MonkeyLearn Blog. Available at: https://monkeylearn.com/blog/data-preprocessing/ [Accessed 4 Nov. 2022].

National Human Genome Research Institute (2022). *Bioinformatics*. [online] Genome.gov. Available at: https://www.genome.gov/genetics-glossary/Bioinformatics. [Accessed 21 Nov. 2022]

O'Brien, J.J., O'Connell, J.D., Paulo, J.A., Thakurta, S., Rose, C.M., Weekes, M.P., Huttlin, E.L. and Gygi, S.P. (2018). Compositional Proteomics: Effects of Spatial Constraints on Protein Quantification Utilizing Isobaric Tags. *Journal of Proteome Research*, [online] 17(1), pp.590–599. doi:10.1021/acs.jproteome.7b00699.

PyShark (2021). *Davies-Bouldin Index for K-Means Clustering Evaluation in Python | Python-bloggers*. [online] Python Bloggers. Available at: https://python-bloggers.com/2021/06/davies-bouldin-index-for-k-means-clustering-evaluation-in-python/ [Accessed 28 Oct. 2022].

San, M. (2016). *Advantages & Disadvantages of k--Means and Hierarchical clustering (Unsupervised Learning) Machine Learning for Language Technology ML4LT (2016)*. [online] Available at: http://santini.se/teaching/ml/2016/Lect_10/10c_UnsupervisedMethods.pdf. [Accessed 1 Nov. 2022].

Shetty, B. (2022). *Curse of Dimensionality*. [online] Built In. Available at: https://builtin.com/data-science/curse-dimensionality [Accessed 30 Oct. 2022].

Sidyakov, M. (2022). *Calinski-Harabasz Index for K-Means Clustering Evaluation using Python*. [online] PyShark. Available at: https://pyshark.com/calinski-harabasz-index-for-k-means-clustering-evaluation-using-python/ [Accessed 28 Oct. 2022].

Sivley, R.M., Dou, X., Meiler, J., Bush, W.S. and Capra, J.A. (2018). Comprehensive Analysis of Constraint on the Spatial Distribution of Missense Variants in Human Protein Structures. *American Journal of Human Genetics*, [online] 102(3), pp.415–426. doi:10.1016/j.ajhg.2018.01.017.

VanderPlas, J. (2019). *In Depth: k-Means Clustering | Python Data Science Handbook*. [online] Github.io. Available at: https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html [Accessed 27 Oct. 2022].

Wei, H. (2020). *How to measure clustering performances when there are no ground truth?* [online] Medium. Available at:

https://medium.com/@haataa/how-to-measure-clustering-performances-when-there-are-no-ground-truth-db027e9a871c#:~:text=The%20Calinski%2DHarabasz%20index%20also [Accessed 13 Nov. 2022].

Whitford, D. (2013). *Proteins: Structure and Function*. [online] *Google Books*. John Wiley & Sons. Available at: https://books.google.com.au/books?id=AnodNhuMAdkC&dq=what+are+proteins&lr=&source=gbs_navlinks_s [Accessed 21 Nov. 2022].

Yufeng (2022). *Understanding Mean Shift Clustering and Implementation with Python*. [online] Medium. Available at: https://towardsdatascience.com/understanding-mean-shift-clustering-and-implementation-with-python-6d5809a2ac40#:~:text=Mean%20shift%20is%20an%20unsupervised [Accessed 22 Nov. 2022].

Zarocostas, J. (2006). Serious birth defects kill at least three million children a year. *BMJ*, [online] 332(7536), p.256.3. doi:10.1136/bmj.332.7536.256-b.

Zhang, S. (2018). *Your Body Acquires Trillions of New Mutations Every Day*. [online] The Atlantic. Available at: https://www.theatlantic.com/science/archive/2018/05/your-body-acquires-trillions-of-new-mutations-every-day/559472/ [Accessed 21 Nov. 2022].

8. Appendix

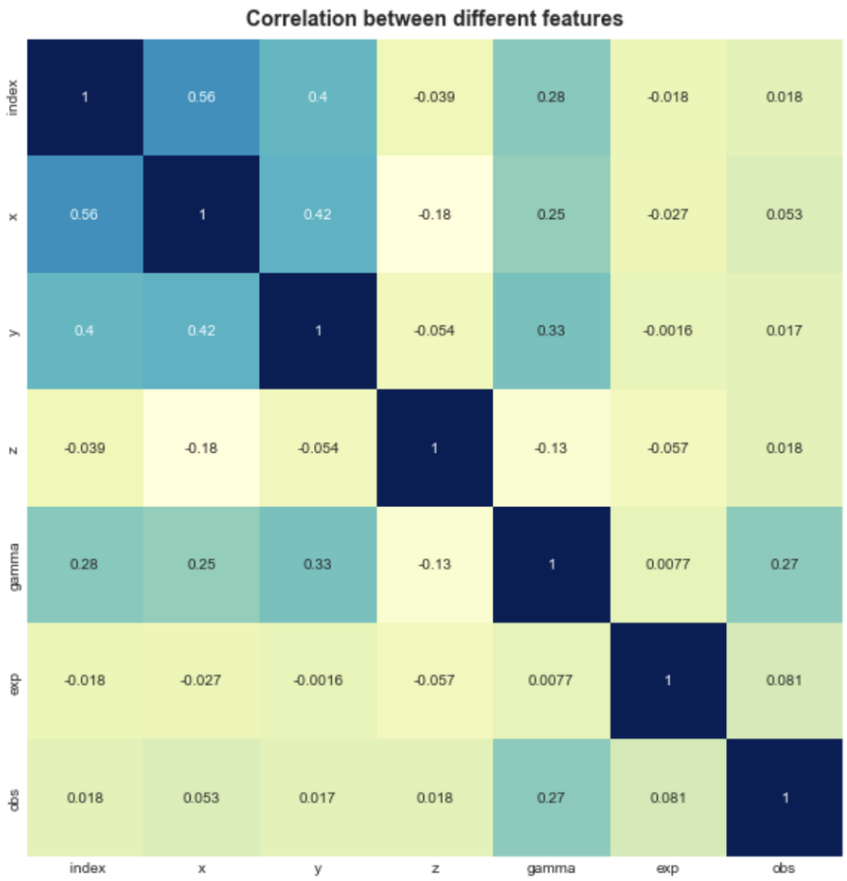
8.1. GitHub repository containing all code used

https://github.com/iamfaiyam/DATA3001-Proteins-Team-2-Repo

8.2. Descriptive statistics of protein O43526

	index	x	y	z	gamma	exp	obs
count	872.000000	872.000000	872.000000	872.000000	872.000000	872.000000	872.000000
mean	436.500000	12.766525	9.374657	-18.326908	0.516791	0.560077	0.291284
std	251.869014	30.937491	24.066195	28.499735	0.282806	0.054157	0.546373
min	1.000000	-44.803000	-52.381000	-96.751000	0.009671	0.385233	0.000000
25%	218.750000	-10.775000	-5.801500	-39.786250	0.272093	0.523433	0.000000
50%	436.500000	8.699000	5.225500	-18.788000	0.513152	0.557733	0.000000
75%	654.250000	30.683750	24.188250	0.469750	0.779246	0.596360	0.000000
max	872.000000	82.551000	66.907000	45.598000	0.997212	0.726519	3.000000

8.3. Correlation matrix of proteins dataset



8.4. 3D plots of proteins

Figure 26: 3d plot of protein P01009

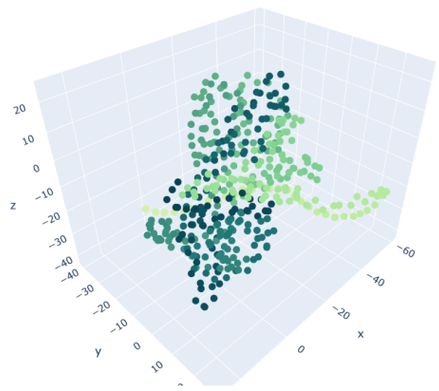


Figure 27: 3d plot of protein P17181

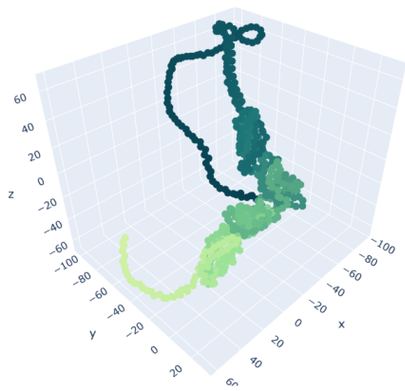


Figure 28: 3d plot of protein P60484

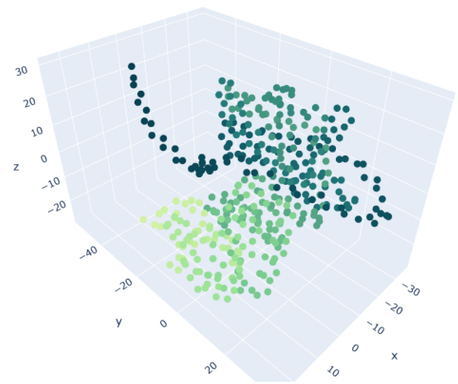


Figure 29: 3d plot of protein P68133

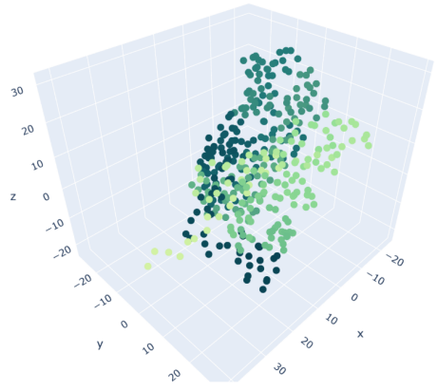


Figure 30: 3d plot of protein Q86VV8

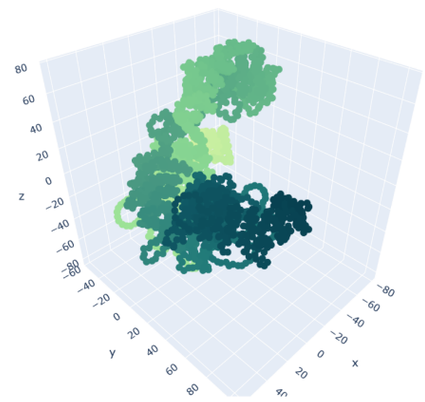
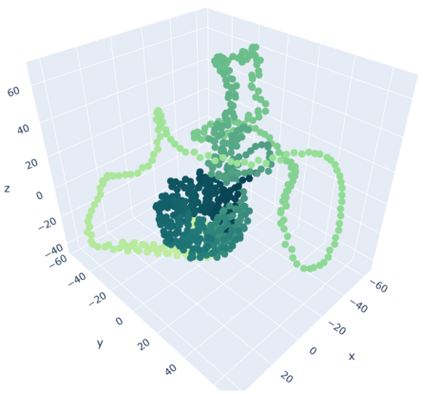


Figure 31: 3d plot of protein Q969H0



8.5. Evaluation metrics on K-Means Clustering

```
At K = 2
Silhouette Coefficient: 0.322
Calinski-Harabasz Index: 458.568
Davies-Bouldin Index: 1.250
---------------------------------
At K = 3
Silhouette Coefficient: 0.316
Calinski-Harabasz Index: 481.443
Davies-Bouldin Index: 1.121
---------------------------------
At K = 4
Silhouette Coefficient: 0.308
Calinski-Harabasz Index: 455.841
Davies-Bouldin Index: 1.101
---------------------------------
At K = 5
Silhouette Coefficient: 0.331
Calinski-Harabasz Index: 438.215
Davies-Bouldin Index: 0.987
---------------------------------
At K = 6
Silhouette Coefficient: 0.324
Calinski-Harabasz Index: 426.495
Davies-Bouldin Index: 1.032
---------------------------------
```

8.6. DBSCAN results for gamma values, minPoint and epsilon value

```
The overall minimum average gamma value of 0.4288679978074875
The number of clusters produced by the DBSCAN clustering 5
List of minimum gamma values: [0.5963903396776727, 0.5057785914437981, 0.3826722554873264, 0.5259469283720811, 0.13355187405655
922]
Count of the data points in each cluster:Counter({0: 553, -1: 265, 3: 26, 1: 14, 2: 14})
Minimum point is 7
Epsilon value: 10.693731107522758
```

8.7. Evaluation metrics for DBSCAN

```
At K = 5
Silhouette Coefficient: -0.185
Calinski-Harabasz Index: 11.472
Davies-Bouldin Index: 4.057
```

8.8. Evaluation metrics for Agglomerative Hierarchical Clustering

```
At K = 2
Silhouette Coefficient: 0.319
Calinski-Harabasz Index: 396.209
Davies-Bouldin Index: 1.288
-----------------------------------
At K = 3
Silhouette Coefficient: 0.254
Calinski-Harabasz Index: 351.430
Davies-Bouldin Index: 1.206
-----------------------------------
At K = 4
Silhouette Coefficient: 0.243
Calinski-Harabasz Index: 334.041
Davies-Bouldin Index: 1.202
-----------------------------------
At K = 5
Silhouette Coefficient: 0.264
Calinski-Harabasz Index: 336.753
Davies-Bouldin Index: 1.222
-----------------------------------
At K = 6
Silhouette Coefficient: 0.282
Calinski-Harabasz Index: 345.717
Davies-Bouldin Index: 1.118
-----------------------------------
```

8.9. Feedback from peer review 1

'The report was well structured, it has a professional and polished format. The logo is a nice touch to the research project. Overall, the report was well conceived with good modelling and background. We also liked the consistent use of diagrams throughout the report, which helped to visualise and add to the contextual understanding of the report (i.e. how the algorithms worked). There was a wide range of models and performance metrics, and detailed elaboration of different modelling approaches. We liked the addition of checking for optimal time complexity, as well as a solid reference length.

It would be ideal to integrate some opinions and outline your experience with the various models and accuracy measures. Comment on their strengths and limitations, and was there a measure more reliable than the other? How does k-means clustering, DBSCAN or AHC allow you to identify constrained zones in the proteins dataset? There was only one protein selected in the report. Assume most of the 10 proteins are like this one, the number of clusters could be different between the different proteins. In addition, it could be explained more clearly and more in depth as green is better than orange, as it got confusing when reading the analysis of results.'

8.10. Feedback from peer review 2

'The draft report well-documented every step in the process to achieve objectives and outlined stage 0 - stage 2 which is clearly explained from the data analysis, methodology to evaluation. By browsing their content, the report is explained precisely and markedly visualised. Throughout the draft report, our team think it clearly showed both sides of expertly researched and written as well as the most distinctive visualisation without exception. However, our group felt that this group lacked a description of the entire data set and did not understand what variables they were subsequently applying to the dataset, which would leave anyone reading the report wondering about the protein and x,y,z axes mentioned later. Hence, we suggest that it would be more comprehensible if descriptions could be added.'

8.11. iCn-3D Web-based Structure Viewer

https://www.ncbi.nlm.nih.gov/Structure/icn3d/full.html