Yeast Protein Localization Sites Clustering

1st Pratyush Gupta
Department of Chemical Engineering
Indian Institute of Technology, Kanpur
pratyushg22@iitk.ac.in

2nd Om Chandrakant Chaudhari
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur
omcc23@iitk.ac.in

3rd Aditya Sharma
Department of Chemistry
Indian Institute of Technology, Kanpur
aditash20@iitk.ac.in

4th Ashutosh Rabia
Department of Mathematics
Indian Institute of Technology, Kanpur
arabia22@iitk.ac.in

***Abstract*—This report presents an analysis of yeast protein localization sites using clustering techniques. The primary objective is to determine the optimal number of clusters and evaluate different clustering models based on silhouette score, Calinski-Harabasz score, and Davies-Bouldin score. Various preprocessing techniques and clustering algorithms were implemented to achieve the best results.**

I. Introduction

The localization of yeast proteins plays a critical role in understanding their biological functions. Given a dataset with 8 features, the task was to identify clusters that best represent different protein localization sites. This was achieved by evaluating multiple clustering algorithms and selecting the optimal model with optimal number of clusters based on performance metrics.

II. Preprocessing

The dataset was preprocessed to ensure effective clustering. The preprocessing steps included:

- Loading and cleaning the dataset by removing irrelevant columns such as "Sequence Name."
- Standardizing the data using StandardScaler to normalize feature distributions.
- Computing the correlation matrix to identify highly correlated features and reduce model complexity.
- Creating box plots to visualize the distribution and detect potential outliers.
- Using DBSCAN to identify and remove outliers before applying clustering algorithms.

III. Model Architecture

A. Finding the Optimal Number of Clusters

To determine the optimal number of clusters, four different evaluation metrics were employed:

- **Elbow Method:** This method evaluates the within-cluster sum of squares (WCSS) as a function of the number of clusters (k). The optimal number of clusters is identified at the "elbow" point, where adding more clusters results in diminishing improvements.

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \tag{1}$$

where C_i represents the i^{th} cluster, μ_i is the centroid of C_i, and x are the data points.

- **Silhouette Score:** This metric measures how similar a data point is to its own cluster compared to other clusters. It is defined as:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2}$$

where $a(i)$ is the average intra-cluster distance, and $b(i)$ is the average nearest-cluster distance for sample i.

- **Calinski-Harabasz Score:** Also known as the Variance Ratio Criterion, it is computed as:

$$CH = \frac{Tr(B_k)}{Tr(W_k)} \times \frac{n - k}{k - 1} \tag{3}$$

where $Tr(B_k)$ is the trace of the between-cluster dispersion matrix, $Tr(W_k)$ is the trace of the within-cluster dispersion matrix, n is the number of samples, and k is the number of clusters.

- **Davies-Bouldin Score:** This index evaluates the average similarity between each cluster and its most similar cluster. It is given by:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{s_i + s_j}{d_{ij}} \right) \tag{4}$$

where s_i is the average distance between points in cluster i, and d_{ij} is the distance between cluster centroids i and j.

B. Clustering Algorithms Used

The following clustering algorithms were implemented:

- **K-Means:** A centroid-based clustering algorithm that minimizes intra-cluster variance:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \tag{5}$$

where C_i is a cluster, and μ_i is its centroid.

- **Agglomerative Clustering:** A hierarchical method that iteratively merges clusters based on linkage criteria such as single-linkage (minimum distance), complete-linkage (maximum distance), or average-linkage.

- **Spectral Clustering:** A graph-based technique that uses the eigenvalues of a similarity matrix to perform dimensionality reduction before clustering.
- **Birch:** A scalable hierarchical clustering method that constructs a clustering feature (CF) tree and incrementally groups data.
- **DBSCAN:** A density-based approach that groups points closely packed together and marks outliers as noise.
- **Gaussian Mixture Model (GMM):** A probabilistic clustering method that models the data distribution as a mixture of multiple Gaussian distributions:

$$p(x) = \sum_{i=1}^{k} \pi_i \mathcal{N}(x|\mu_i, \Sigma_i) \tag{6}$$

where π_i are the mixture weights, and $\mathcal{N}(x|\mu_i, \Sigma_i)$ represents the Gaussian distribution with mean μ_i and covariance Σ_i.

IV. Results and Discussion

The models were evaluated based on clustering metrics, and the optimal number of clusters was determined. The best clustering method was selected by considering:

Silhouette Score: Measures how well-separated the clusters are.

Calinski-Harabasz Score: Evaluates the ratio of between-cluster dispersion to within-cluster dispersion.

Davies-Bouldin Score: Determines the average similarity between clusters.

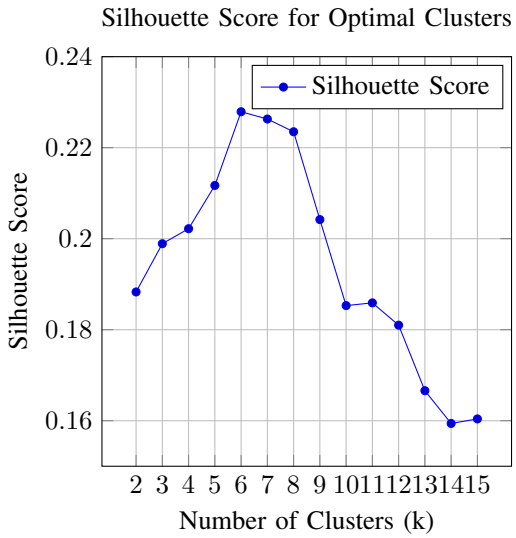


Fig. 1. Silhouette Score for Different Clusters.

The final model selection was based on optimizing clustering performance by maximizing the **Silhouette Score** and **Calinski-Harabasz Score**, while minimizing the **Davies-Bouldin Score**. After evaluating various clustering methods, the results indicated that an optimal cluster count of **8** provided the best clustering performance. Among the tested algorithms, **K-Means** achieved the highest performance, demonstrating superior cluster cohesion and separation.

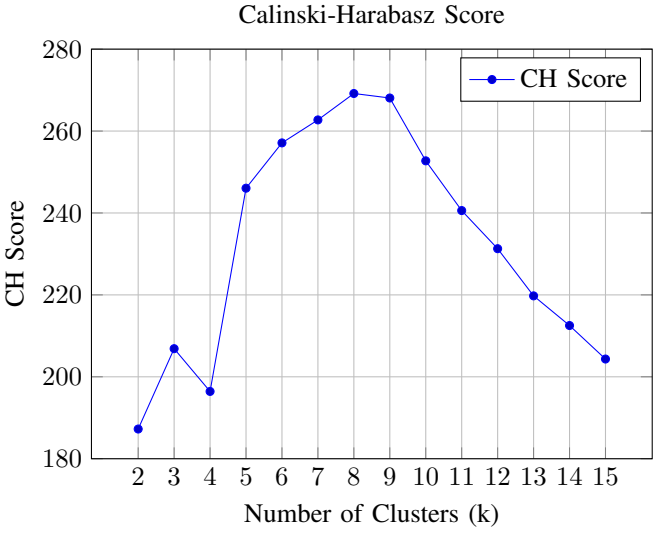
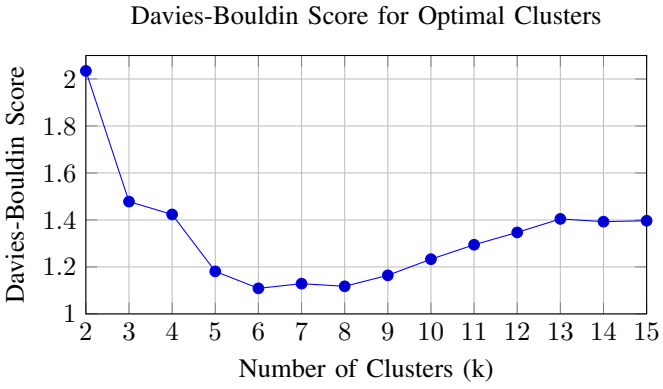


Fig. 2. Calinski-Harabasz Score for Different Clusters.



Method	Silhouette Score	Calinski-Harabasz Score	Davies-Bouldin Score
KMeans	0.2235	269.1713	1.1175
Agglomerative-Single	0.4838	65.9631	0.5655
Agglomerative-Average	0.3439	72.5807	0.6115
Agglomerative-Complete	0.2132	144.4589	1.1414
BIRCH	0.1771	222.2297	1.2425
Gaussian Mixture	-0.0383	112.4926	2.4976
DBSCAN	-0.3718	1.9526	2.2676

TABLE I
CLUSTERING PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS WITH 8 CLUSTERS.