Lab 1: Clustering

Alejandro Silva Rodríguez

Marta Cuevas Rodríguez

Aprendizaje Computacional
Universidad de Málaga

Septiembre 2024

Índice

1. Introduction

The process of **sporulation in yeast** is a well-established model for studying cellular differentiation and gene regulation. Sporulation involves a series of highly regulated biological stages during which the yeast cell transitions into a spore, primarily in response to nutrient deprivation. Gene expression in yeast during sporulation is characterized by distinct temporal patterns, making it an ideal candidate for clustering analysis. Through clustering, genes with similar expression profiles can be grouped, aiding in the identification of genes that may participate in similar biological functions or regulatory pathways.

With the advent of **microarray technology**, the ability to measure the expression levels of thousands of genes simultaneously across different time points has significantly advanced. This vast amount of data requires effective computational tools for analysis. One such tool is **clustering**, which groups genes based on the similarity of their expression profiles. In this context, **k-Means clustering** has emerged as a widely used technique due to its simplicity and effectiveness. The algorithm attempts to partition genes into k clusters by minimizing the variance within each cluster, leading to groups of genes that exhibit similar temporal expression patterns during sporulation.

In this project, we aim to evaluate the performance of **k-Means clustering on the sporulation dataset** of budding yeast, comparing the results to those presented in by Datta and Datta (2003)[2], which explores various clustering methods including hierarchical clustering and Diana. The primary objective is to assess the effectiveness of k-Means in clustering genes during sporulation, and to analyze how it compares to more complex methods discussed in the literature.

2. Objectives

The main objective of this project is to **evaluate the performance of the k-Means clustering algorithm** when applied to the sporulation dataset of yeast, which contains gene expression profiles measured across multiple time points. By clustering these genes, the aim is to **identify groups of genes** that exhibit similar expression patterns throughout the sporulation process. To assess the effectiveness of k-Means, the results will be compared to those obtained in Datta and Datta's (2003) study [2], which evaluated various clustering techniques, including hierarchical clustering and divisive clustering (Diana). Additionally, metrics such as the silhouette score will be used to quantify the quality of the clustering results. Through this comparison, the project seeks to determine the strengths and limitations of k-Means in clustering biological data and to explore its applicability in gene expression analysis during yeast sporulation.

3. Methodology and Results

We begin by preparing the environment for data processing and clustering using the Sporulation Yeast Dataset, which contains gene expression data measured at 7 distinct time points during the sporulation process. **Each row in the dataset represents a gene**, and the **columns correspond to the expression levels at different time intervals**. To focus on the relevant data, we exclude the mean and variance rows from the dataset. We designate the gene names as the Y-axis (or labels) and the time points as the X-axis (features), allowing us to analyze how gene expression levels vary over time. This preprocessing step ensures that the data is structured correctly for subsequent clustering analysis.

The next step is to **normalize** the gene expression data using Z-score normalization. This method adjusts the values of each gene's expression levels to have a mean of 0 and a standard deviation of 1 across the time points. By doing this, we ensure that the expression data for each gene is comparable, preventing any gene with higher baseline expression levels from dominating the clustering process.

Now we employ the Elbow Method to identify the optimal number of clusters, k, for the k-Means algorithm. The Elbow Method is a well-established technique used to determine the appropriate number of clusters by evaluating the within-cluster sum of squares (also known as inertia) for a range of cluster values.

Listing 1: Implementation of Elbow Method to the dataset

```
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer

model = KMeans()
visualizer = KElbowVisualizer(model, k=(2,9)) # a range of k values from 2 to 9

visualizer.fit(x)         # Fit the data to the visualizer
visualizer.show()         # Finalize and render the figure
```

By using the KElbowVisualizer we assess the inertia for k values between 2 and 9 as is shown in the listing 1. The visualizer generates the figure 1.

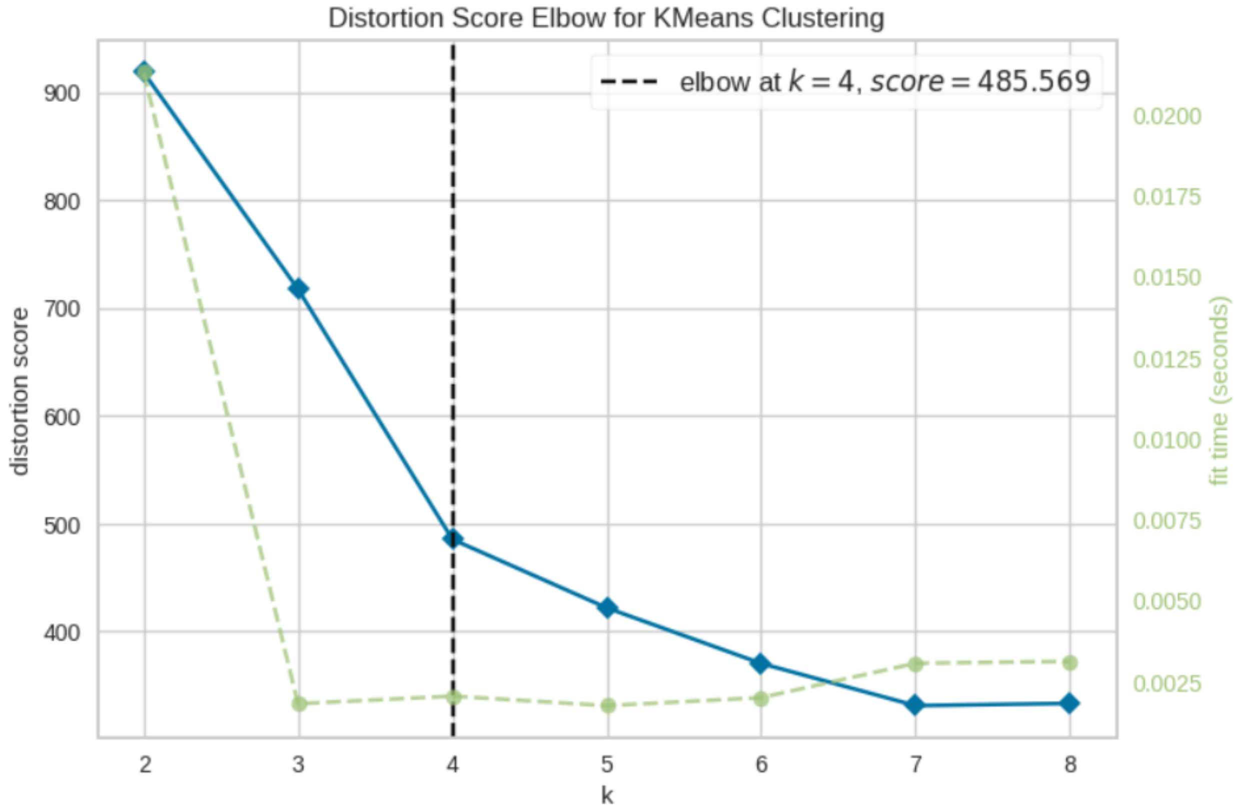

Figura 1: Distortion Score Elbow for KMeans Clustering

The 'elbow' in 1, which represents the point where the reduction in inertia starts to slow down, indicates the optimal number of clusters. In this case, the analysis reveals that **k=4** is the optimal number of clusters, suggesting that partitioning the dataset into 4 clusters strikes a balance between minimizing within-cluster variance and avoiding overfitting.

To give more information, **Principal Component Analysis (PCA)** is applied to the gene expression dataset to reduce its dimensionality from the original high-dimensional space to two dimensions. By creating an instance of the PCA class with n components=2, the code in the listing 2 captures the most significant variance in the data while transforming it into a lower-dimensional representation.

Listing 2: Application of PCA method

```
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns

pca = PCA(n_components=2)
x_pca = pca.fit_transform(x)


plt.figure(2, figsize=(8, 6))
plt.clf()
plt.scatter(x_pca[:, 0], x_pca[:, 1], cmap=plt.cm.Set1, edgecolor="k")
# plt.scatter(x_pca[:, 0], x_pca[:, 1], c=y, cmap=plt.cm.Set1, edgecolor="k")

plt.xlabel("First␣principal␣component")
plt.ylabel("Second␣principal␣component")
```

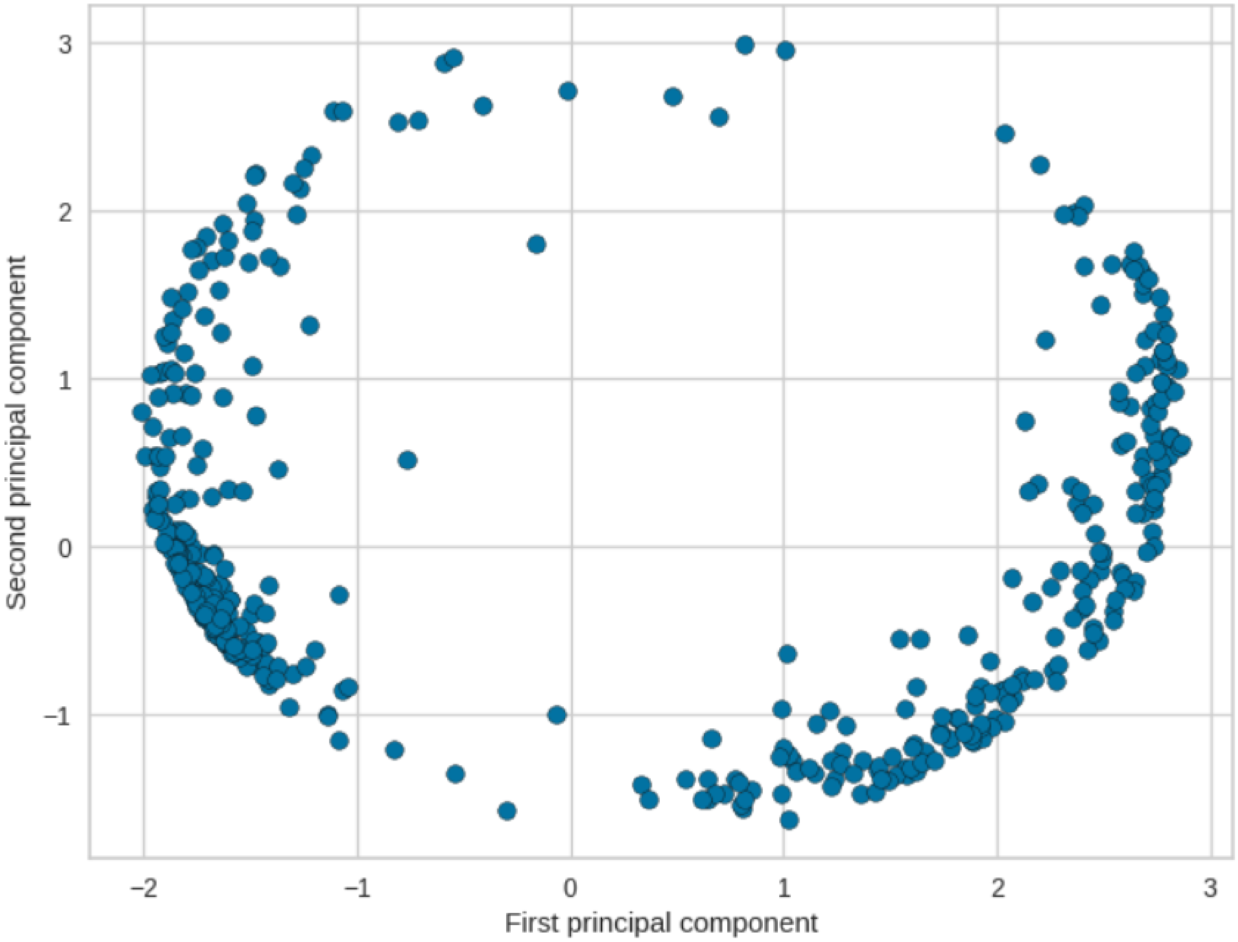

Figura 2: PCA Visualization of Gene Expression Profiles During Sporulation

The resulting two principal components are then visualized in 2 using a scatter plot, where the x-axis represents the first principal component and the y-axis represents the second principal component. This visualization allows for a clearer understanding of the data's structure, revealing potential clusters and relationships among gene expression profiles during the sporulation process. Overall, PCA facilitates a more straightforward interpretation of complex datasets, making it easier to analyze patterns in gene expression.

poner codigo de algunas cosas

imagenes y discusion con respecto a el paper

quedamos en que con nuestros datos el k=2 porque no tenemos los mismos datos que los demas. los del paper lo comparan con informacion externa que tiene mas sentido pero no es comparable con los datos de los otros ni nada.

4. Acceso al Repositorio

Toda la información adicional, incluyendo el código fuente y la documentación completa de este proyecto, está disponible en el repositorio de GitHub [1].

Referencias

[1] Alex Silva. Practica1_almacenes_de_datos. `https://github.com/AlexSilvaa9/Practica1\_almacenes\_de\_datos`, 2024. Último acceso: 1 octubre 2024.

[2] Susmita Datta and Somnath Datta. Comparison and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, 19(4):459–466, 2003. `http://bioinformatics.oxfordjournals.org/content/19/4/459`, 2003.