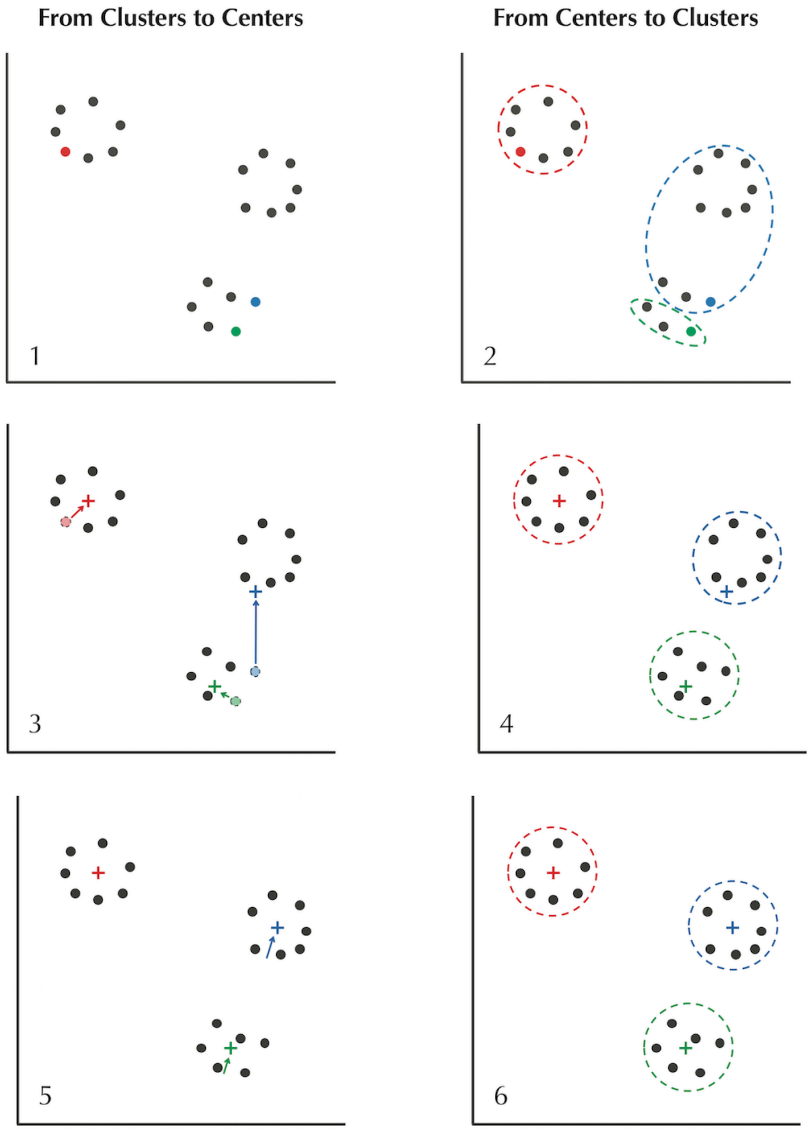
CS4054
Bioinformatics

Spring 2025

Rushda Muneer

The Lloyd Algorithm

- The Lloyd algorithm is one of the most popular clustering heuristics for the *k*-Means Clustering Problem.
- It first chooses *k* arbitrary distinct points *Centers* from *Data* as centers and then iteratively performs the following two steps:
  - **Centers to Clusters:** After centers have been selected, assign each data point to the cluster corresponding to its nearest center; ties are broken arbitrarily.
  - **Clusters to Centers:** After data points have been assigned to clusters, assign each cluster's center of gravity to be the cluster's new center.

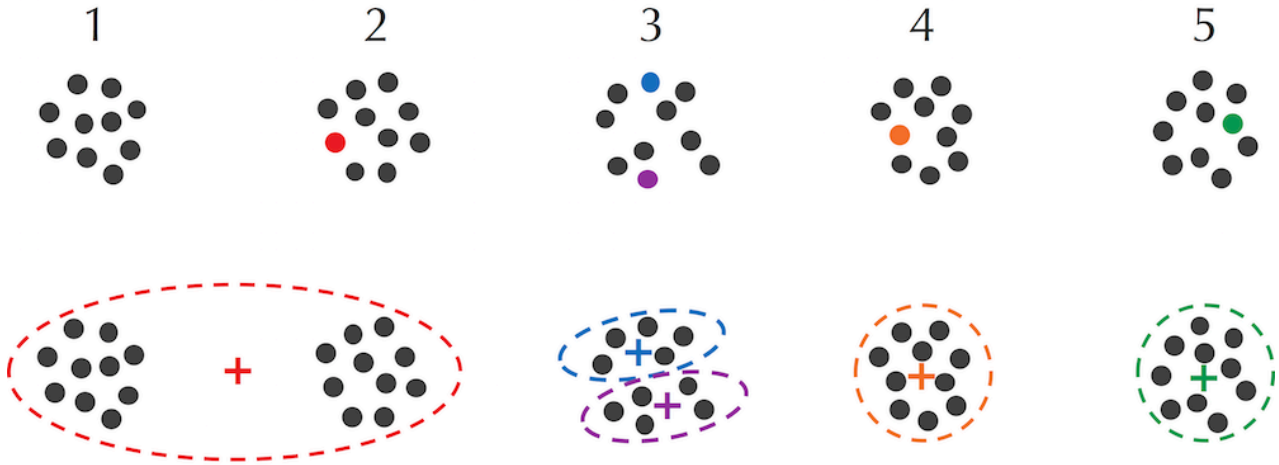

From Clusters to Centers
From Centers to Clusters
1
2
3
4
5
6

Rules for convergence

- If the Lloyd algorithm has not converged, the squared error distortion must decrease in any step, according to the following reasoning:
- In a “Centers to Clusters” step, if a data point is assigned to a new center, then **this point must be closer to the new center than its previous center**. Thus, **the squared error distortion must decrease**.
- In a “Clusters to Center” step, if a center is updated as a cluster’s center of gravity, then **the new center is the only point minimizing the squared error distortion for the points in its cluster**. Thus, **the squared error distortion must decrease.**

- **Exercise Break:** Run the Lloyd algorithm for the set of four one-dimensional data points {0, 1, 1.9, 3} with the two initial centers {1, 3}.

Initializing the Lloyd algorithm



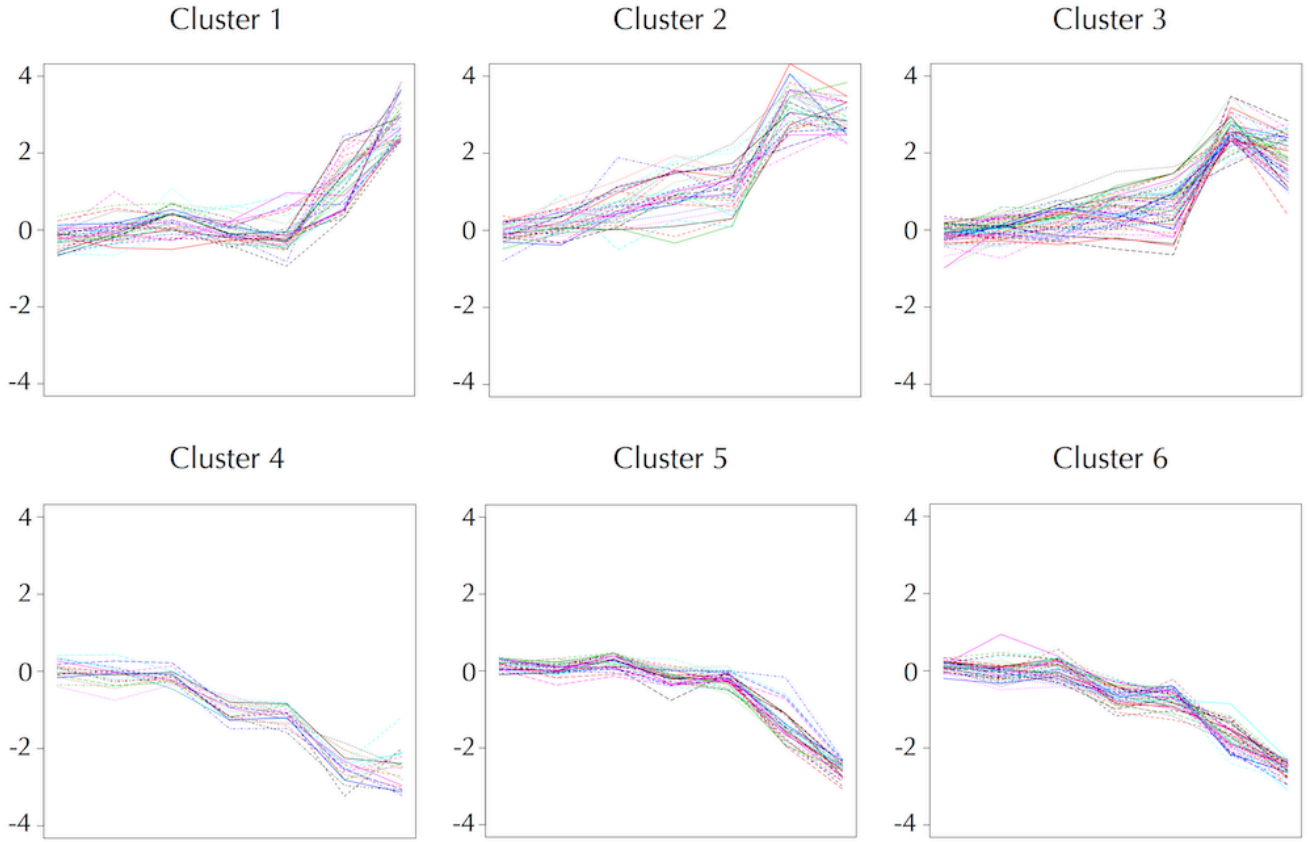
- If we do not pay attention to the Lloyd algorithm's initialization step
- (Top) Five clumps of ten points in two-dimensional space.
- The Lloyd algorithm is initialized so that clump 1 contains no centers, clump 3 contains two centers (blue and purple), and each of the other three clumps contains one center (red, orange, and green).
- (Bottom) Points are clustered according to the center to which they are assigned. The Lloyd algorithm has combined the points in clumps 1 and 2 into a single cluster and split clump 3 into two clusters.

k-means++ Initializer

- We have thus far not paid much attention to how initial centers are chosen in the Lloyd algorithm, which selects them randomly.
- Similarly to **Farthest First Traversal**, **k-Means++Initializer** picks k centers one at a time, but instead of choosing the point farthest from those picked so far, **it chooses each point at random in such a way that distant points are more likely to be chosen than nearby points.**
- Specifically, **the probability of selecting a center *DataPoint* from *Data* is proportional to the squared distance of *DataPoint* from the centers already chosen**, i.e., to $d(DataPoint, Centers)^2$.

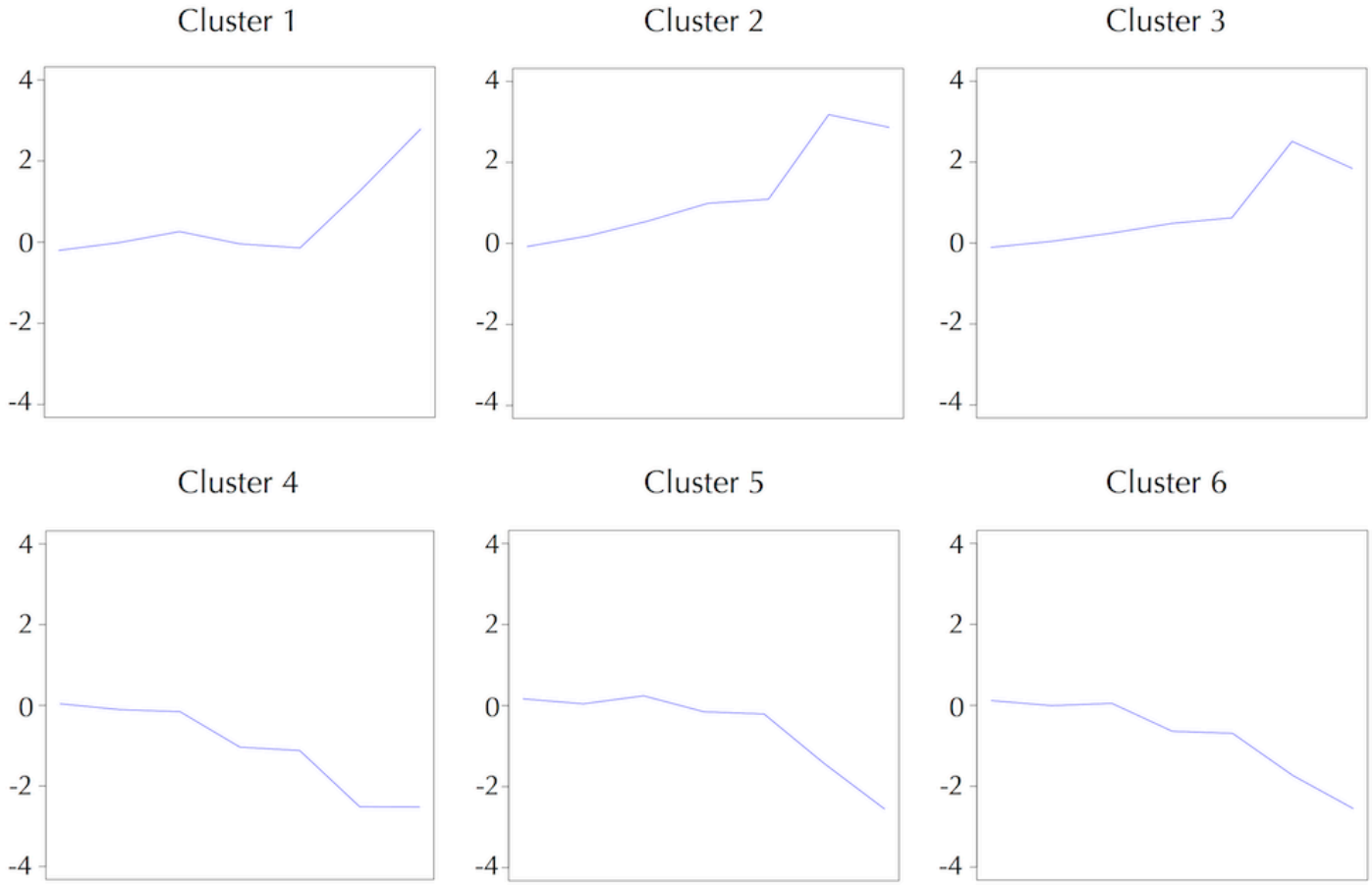
Clustering Genes Implicated in the Diauxic Shift

Since selecting the most biologically relevant value of *k* can be challenging, we will (somewhat arbitrarily) choose to cluster the 230 yeast genes into six clusters



- Applying the Lloyd algorithm (with *k* = 6) to the abridged yeast dataset containing 230 genes results in six clusters containing 37, 36, 58, 19, 36, and 44 genes

Clustering Genes Implicated in the Diauxic Shift

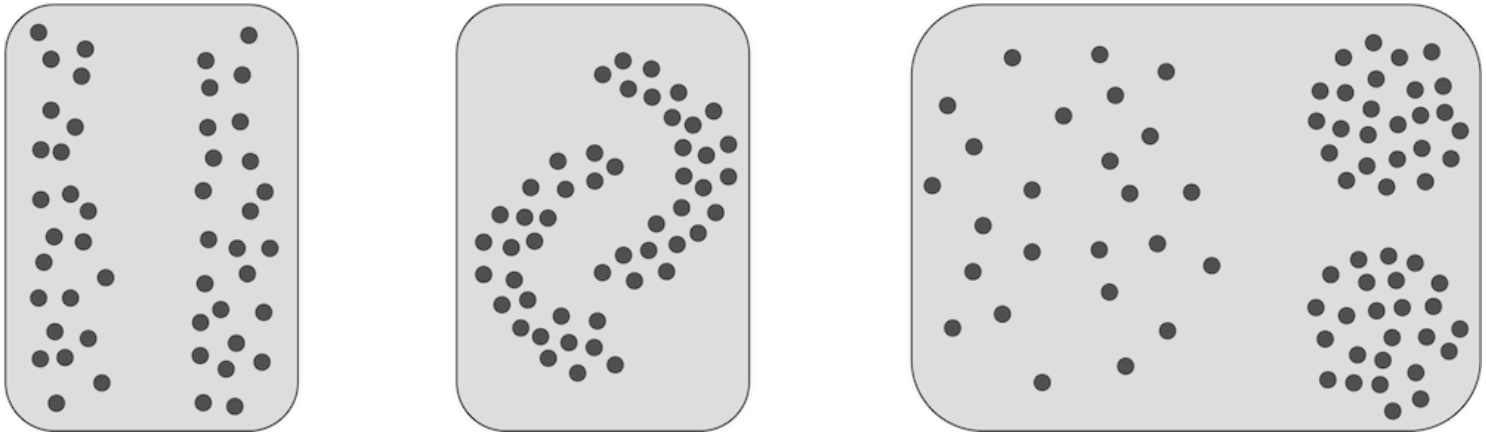


Averaging the expression vectors in each plot reveals six different types of regulatory behavior.

- To reveal the patterns of the expression vectors in each cluster, we average these vectors

Limitations of k-means Clustering

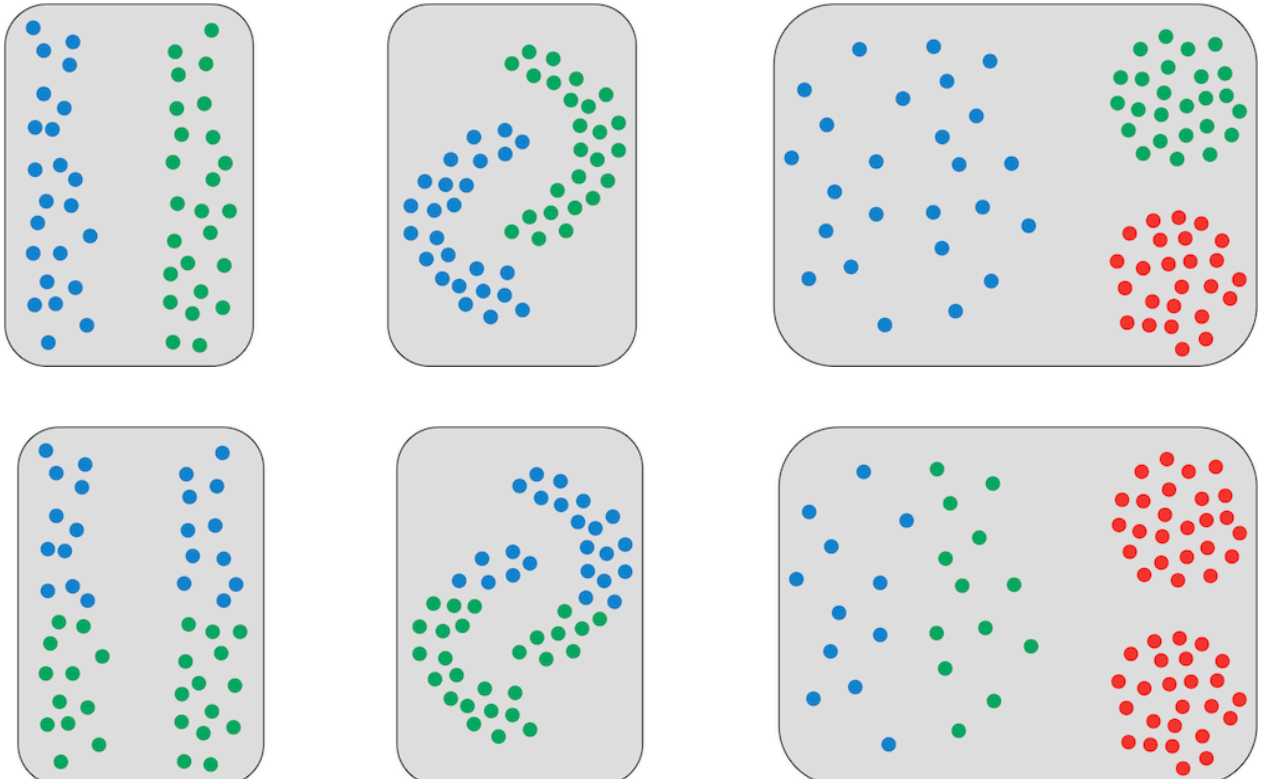
- After seeing the Lloyd algorithm in action, it may seem that clustering is easy. If you think so, consider the following question.
- **STOP and Think:** How would you cluster the points in the figure below?



k = 2 (left and middle) and *k* = 3 (right).

Limitations of k-means Clustering

- In the case of challenging clustering problems, the Lloyd algorithm sometimes fails to identify what may seem like obvious clusters



The human eye (top) and the Lloyd algorithm (bottom) often disagree in the case of elongated clusters (left), clusters with non-globular shapes (middle), and clusters with widely different data point densities (right).

Introduction to distance-based clustering

- Biologists do not always analyze the $n \times m$ gene expression matrix directly.
- Instead, they sometimes first transform this matrix into an $n \times n$ **distance matrix** D, where $D_{i,j}$ indicates the distance between the expression vectors for genes i and j

	1 hr	2 hr	3 hr
g_1	10.0	8.0	10.0
g_2	10.0	0.0	9.0
g_3	4.0	8.5	3.0
g_4	9.5	0.5	8.5
g_5	4.5	8.5	2.5
g_6	10.5	9.0	12.0
g_7	5.0	8.5	11.0
g_8	3.7	8.7	2.0
g_9	9.7	2.0	9.0
g_{10}	10.2	1.0	9.2

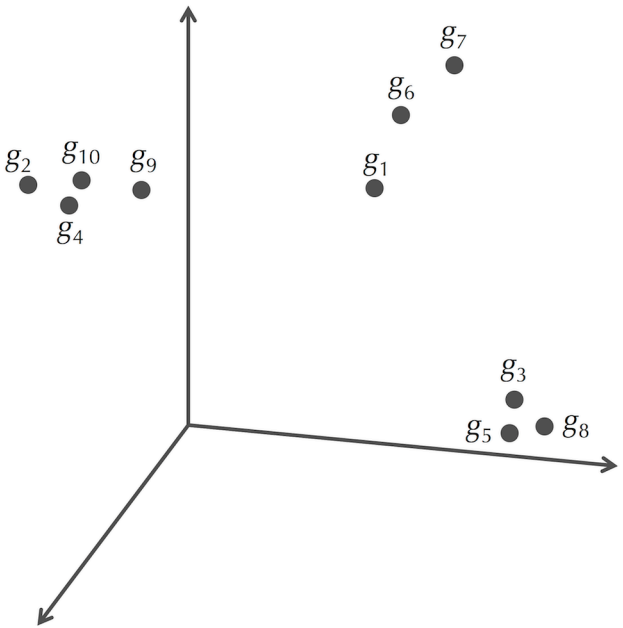
Expression Matrix

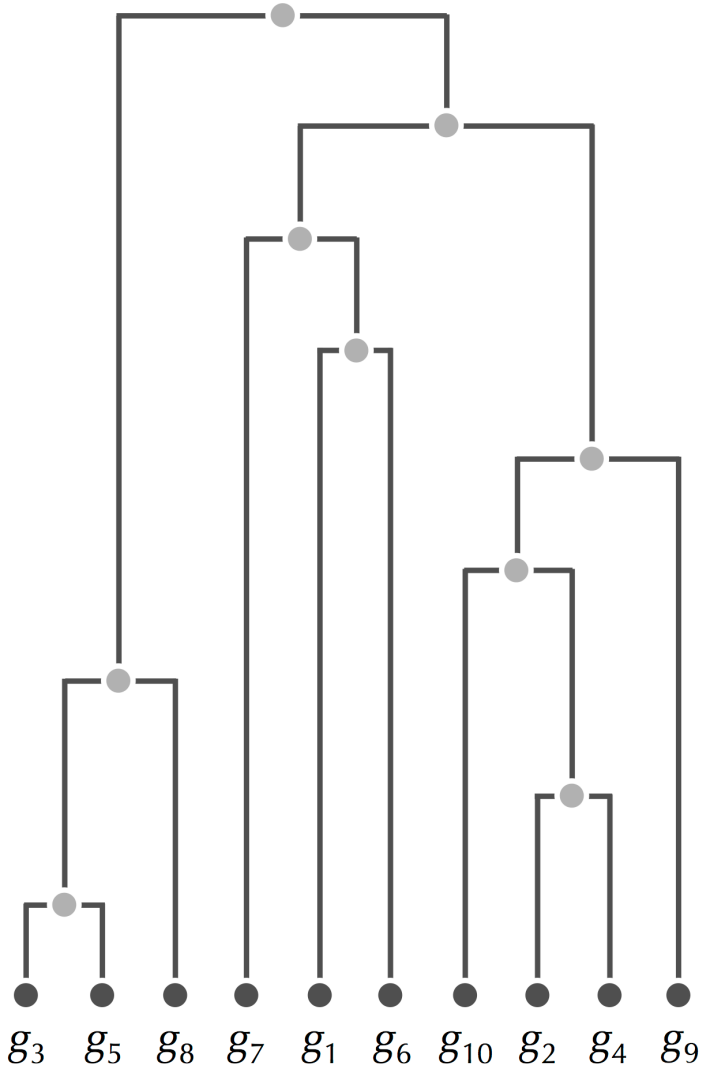
	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0

Distance Matrix

Hierarchical Clustering

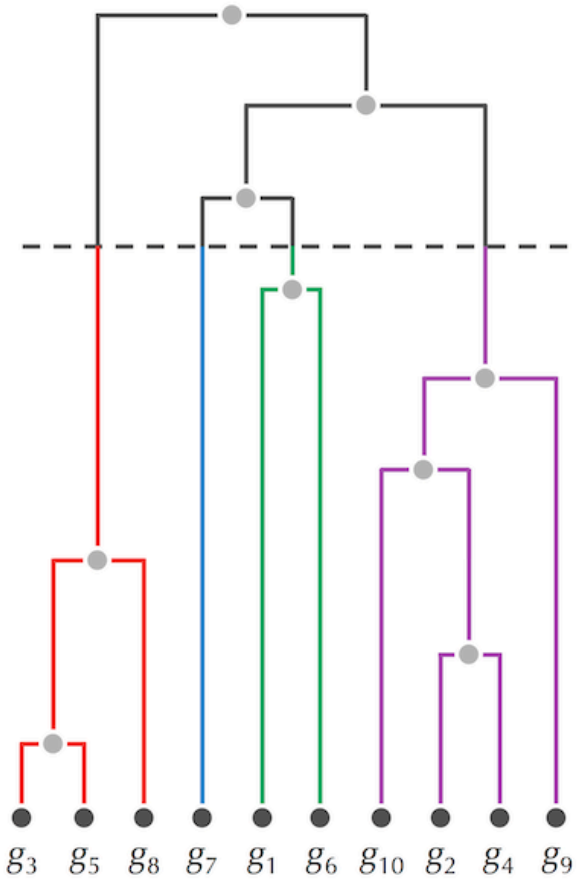
- Previously, we assumed that we were working with a fixed number of clusters *k*.
- But in practice, clusters often have subclusters, which have subsubclusters, and so on.
- To capture this cluster stratification, the hierarchical clustering algorithm uses an $n \times n$ distance matrix D to organize n data points into a tree

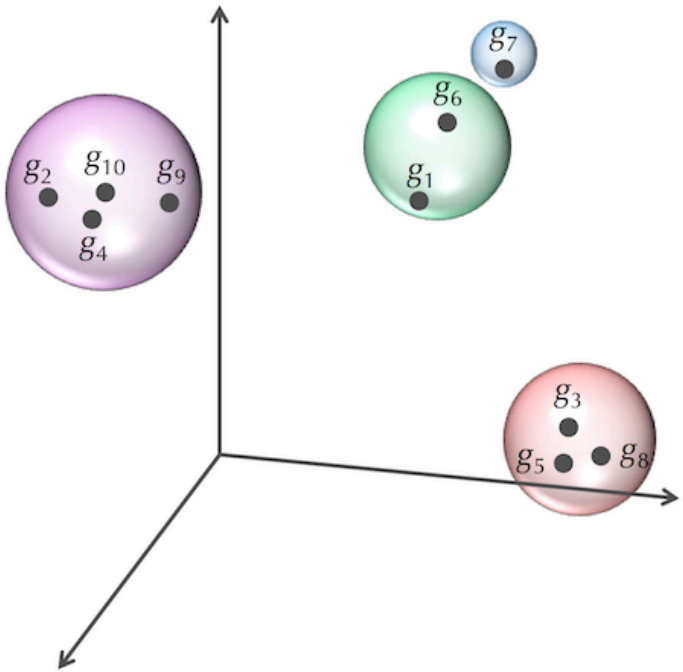




Hierarchical Clustering

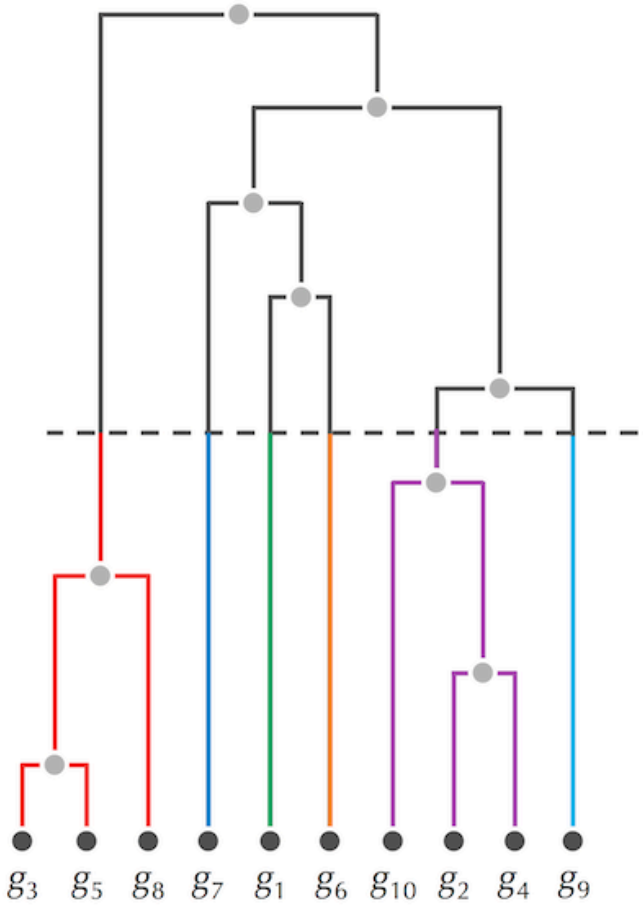
- A horizontal line crossing the tree in *i* places divides the *n* genes into *i* clusters.



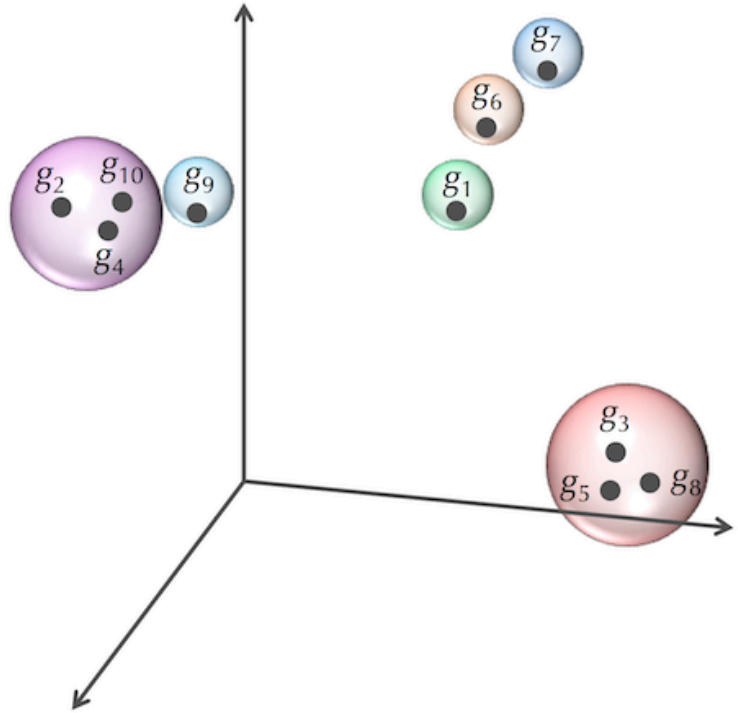


A tree with *n* leaves imposes *n* different ways of partitioning the data into clusters. The horizontal line through the tree (left) crosses the data in four places and partitions the data into four clusters (right).

Hierarchical Clustering



- The same tree with a different horizontal line (left) partitions the data into six clusters (right).



Hierarchical Clustering

- One commonly used approach defines the distance between clusters C_1 and C_2 as the smallest distance between any pair of elements from these clusters (single link)

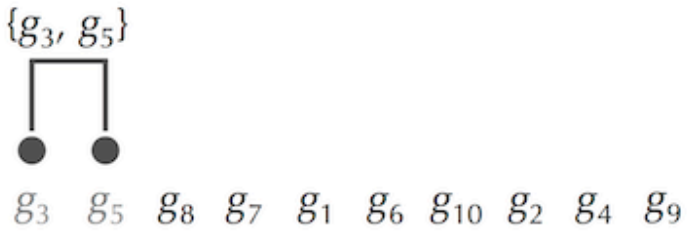
$$D_{\text{min}}(C_1,C_2) = \min_{\text{all points } i \text{ in cluster } C_1,\ \text{all points } j \text{ in cluster } C_2} D_{i,j} \ .$$

- Another distance function uses the average distance between elements in two clusters (average link)

$$D_{\text{avg}}(C_1, C_2) = \frac{\sum_{\text{all points } i \text{ in cluster } C_1} \sum_{\text{all points } j \text{ in cluster } C_2} D_{i,j}}{|C_1| \cdot |C_2|}$$

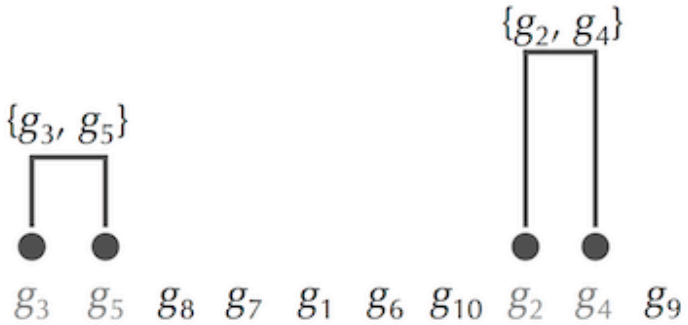
- Combining the closet genes based on the distance matrix can yield hierarchical clustering.

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	**0.7**	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0



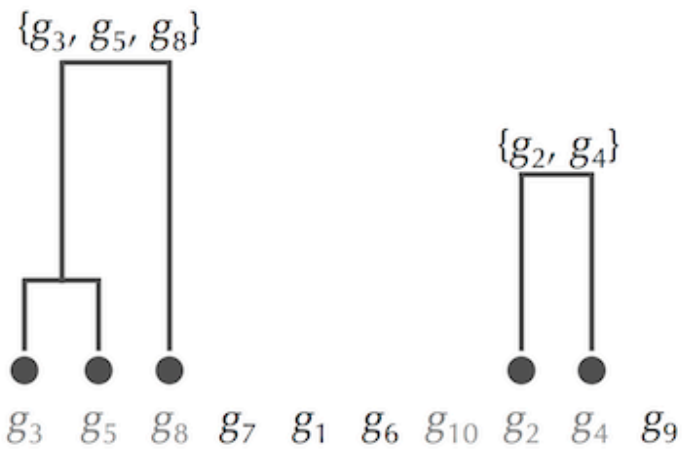
- Combining the closet genes based on the distance matrix can yield hierarchical clustering.

	g_1	g_2	g_3, g_5	g_4	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	**0.9**	9.5	10.1	12.8	2.0	1.0
g_3, g_5	9.2	12.0	0.0	11.2	11.1	8.1	1.0	10.5	11.5
g_4	7.7	0.9	11.2	0.0	9.2	9.5	12.0	1.6	1.1
g_6	2.3	9.5	11.1	9.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.0	12.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	8.5	9.3	12.4	1.1	0.0



- Combining the closet genes based on the distance matrix can yield hierarchical clustering.

	g_1	g_2, g_4	g_3, g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	7.7	9.2	2.3	5.1	10.2	6.1	7.0
g_2, g_4	7.7	0.0	11.2	9.2	9.5	12.0	1.6	1.0
g_3, g_5	9.2	11.2	0.0	11.1	8.1	**1.0**	10.5	11.5
g_6	2.3	9.2	11.1	0.0	5.6	12.1	7.7	8.5
g_7	5.1	9.5	8.1	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	1.6	10.5	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	8.5	9.3	12.4	1.1	0.0



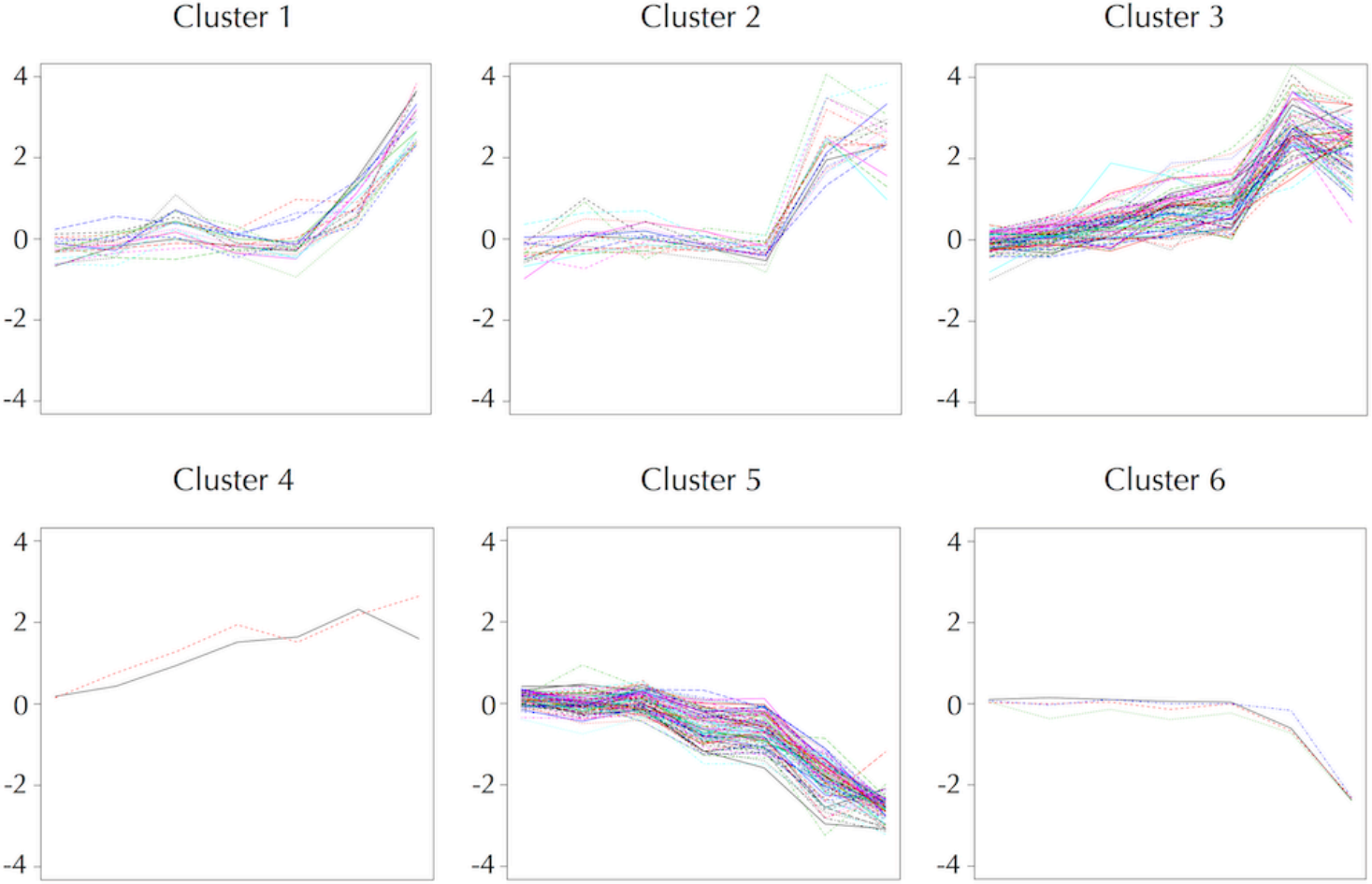
- **Exercise Break:** Apply **Hierarchical Clustering** to the distance matrix using D_{min} and D_{avg} .

$$D_{\text{avg}}(C_1, C_2) = \frac{\sum_{\text{all points } i \text{ in cluster } C_1} \sum_{\text{all points } j \text{ in cluster } C_2} D_{i,j}}{|C_1| \cdot |C_2|}$$

$D_{\text{min}}(C_1,C_2) = \min_{\text{all points } i \text{ in cluster } C_1, \text{ all points } j \text{ in cluster } C_2} D_{i,j}$.

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0

Analyzing the diauxic shift with hierarchical clustering



- The figure visualizes expression vectors for each of the six clusters obtained after applying **Hierarchical Clustering** (using D_{avg}) to the yeast dataset.

Their averages are shown below