Bioinformatics: Gene Expression Analysis-I

Swakkhar Shatabda

Department of Computer Science and Engineering
BRAC University

Book Reference

Bioinformatics Algorithms, An Active Learning Approach , Vol 2, Chapter 8

BRAC UNIVERSITY
Inspiring Excellence

Machine Learning

A computer program is said to learn from **experience E** with respect to some class of **task T** and **performance measure P**, if its performance at task T as measured by P improves with experience E.

- Task T to be performed
  - Classification, Regression, Transcription, Translation, Structured Output, Anomaly Detection, Synthesis, Imputation, Denoising
- Measured by Performance Measure P
  - accuracy, ari, mae, R^2, psnr, etc.
- Trained on Experience E (Training Data)
  - sequence, structure, physico-chemical properties, gene-count, etc.

BRAC UNIVERSITY
Inspiring Excellence

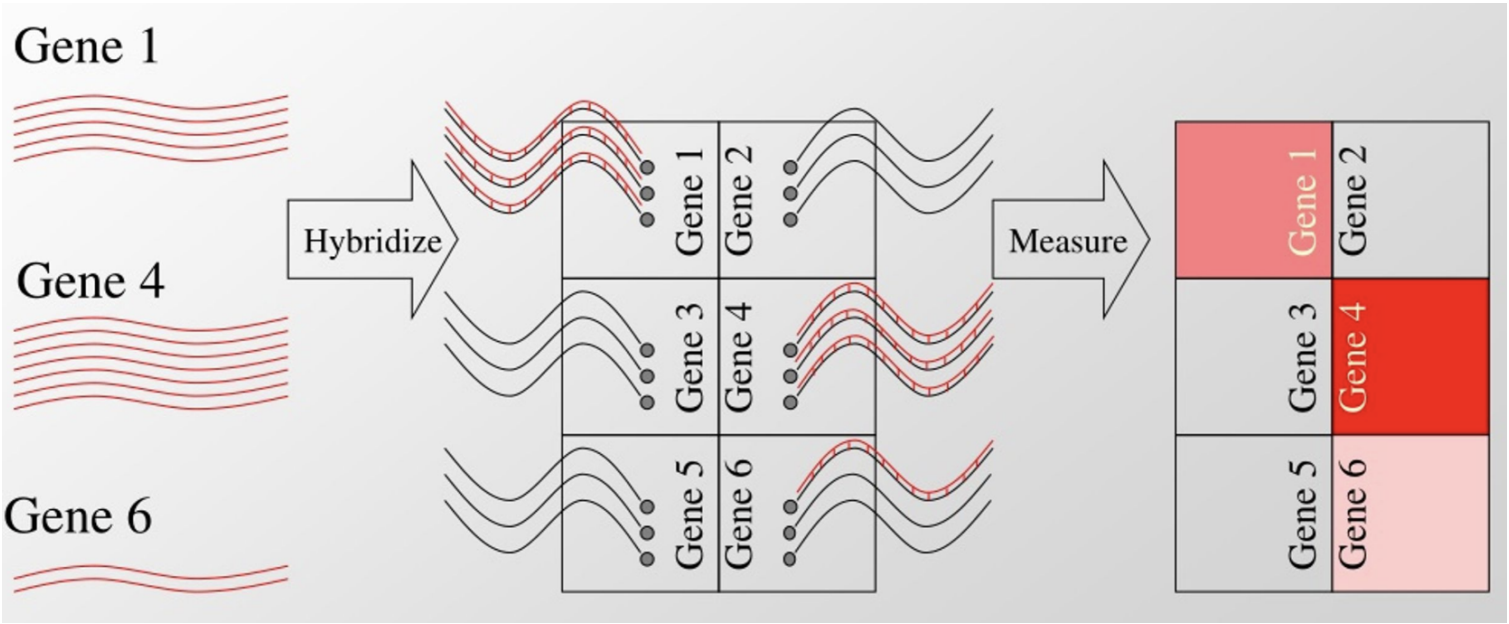
Methods for Measuring Gene Expressions

- The most intuitive way to investigate a certain phenotype is to measure the expression levels of functional proteins present at a given time in the cell.
- Measuring the concentration of proteins can be difficult, due to their varying locations, modifications, and contexts in which they are found, as well as due to the incompleteness of the proteome.
- mRNA expression levels, however, are easier to measure, and are often a good approximation.
- Two techniques:
  - Expression Microarrays
  - RNA-seq
- In micro-array experiments, short segments of DNA, known as probes, are attached to a solid surface, commonly known as a gene chip. RNA population of interest, which has been taken from a cell, is reverse transcribed to cDNA (complementary DNA)

BRAC UNIVERSITY
Inspiring Excellence

Expression Microarrays

- The resulting DNA has more complementarity to the DNA on the slide than the RNA. The cDNA is than washed over the chip and the resulting hybridization triggers the probes to fluoresce.

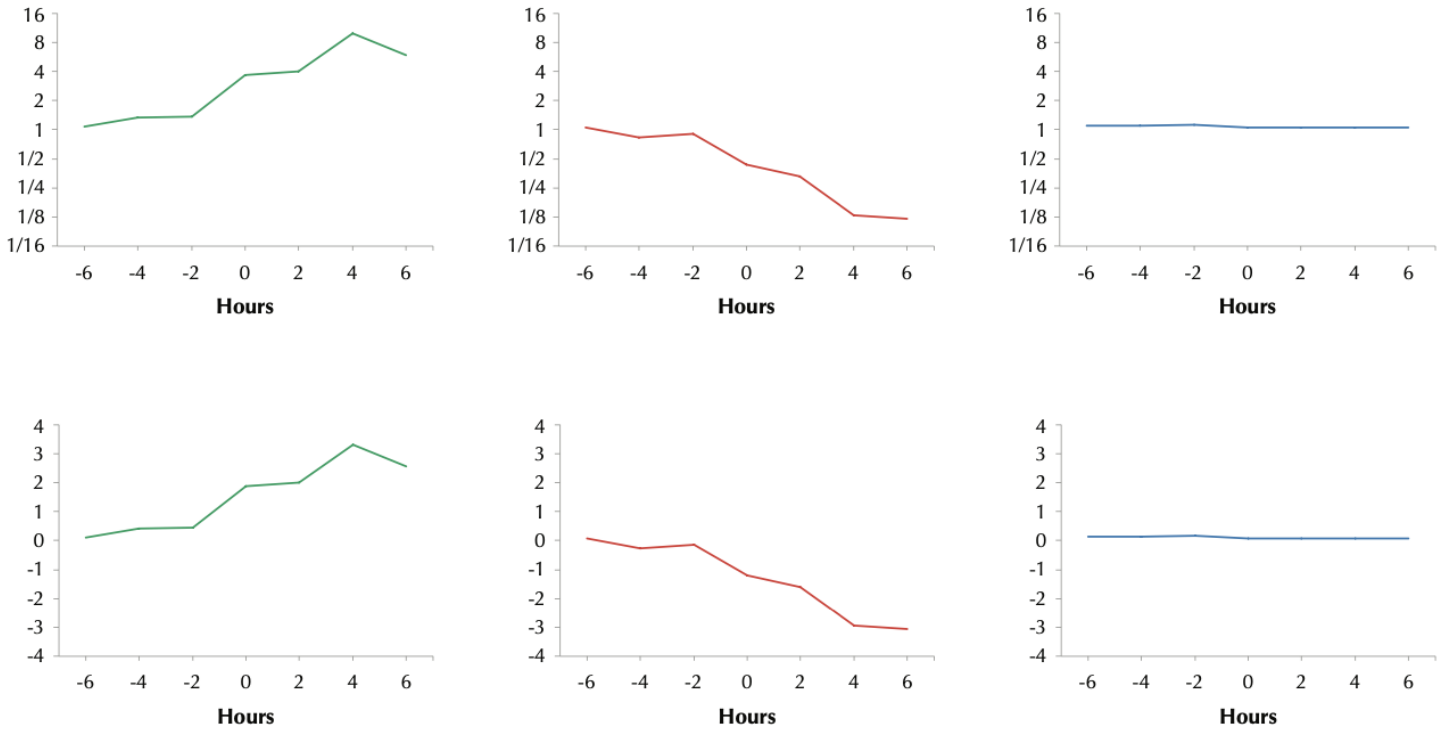


- It cannot distinguish mRNA isoforms, it cannot analyze on the sequence, or digital level, it can only measure known transcripts, and the expression measurements become less reliable for highly saturated transcript levels.

BRAC UNIVERSITY
Inspiring Excellence

Expression Array

- Monitor n genes at m time checkpoints or conditions, resulting in an $n \times m$ gene expression matrix E, where $E_{i,j}$ is a number representing the expression level of gene i at checkpoint/condition j.
- The i-th row of E is called the expression vector of gene i.
- Just by looking at the gene expression vectors, you would observe different patterns of gene behavior with respect to the checkpoints/conditions.



Expression Array

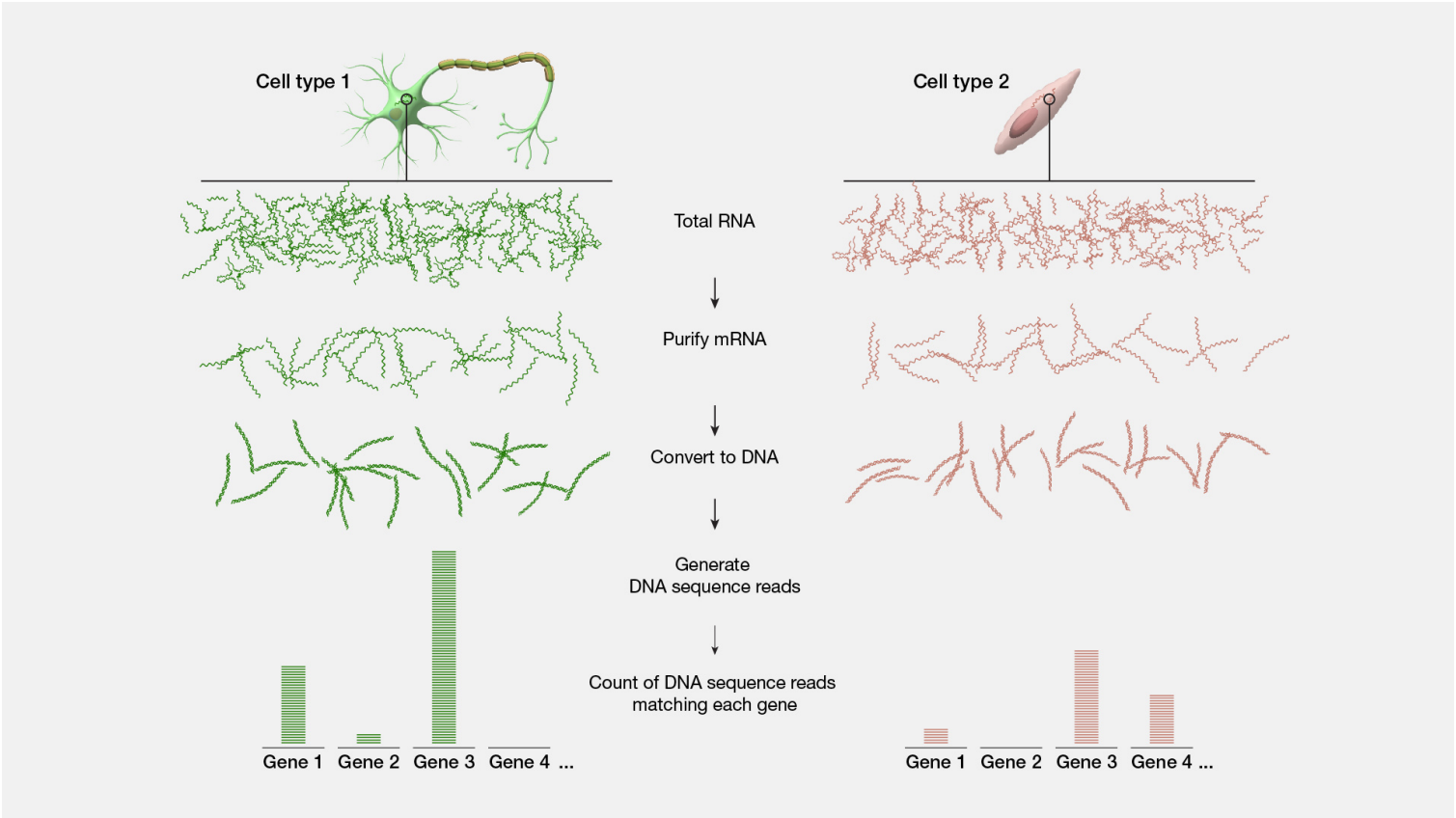
Gene	Expression Vector						
YLR361C	0.14	0.03	-0.06	0.07	-0.01	-0.06	-0.01
YMR290C	0.12	-0.23	-0.24	-1.16	-1.40	-2.67	-3.00
YNR065C	-0.10	-0.14	-0.03	-0.06	-0.07	-0.14	-0.04
YGR043C	-0.43	-0.73	-0.06	-0.11	-0.16	3.47	2.64
YLR258W	0.11	0.43	0.45	1.89	2.00	3.32	2.56
YPL012W	0.09	-0.28	-0.15	-1.18	-1.59	-2.96	-3.08
YNL141W	-0.16	-0.04	-0.07	-1.26	-1.20	-2.82	-3.13
YJL028W	-0.28	-0.23	-0.19	-0.19	-0.32	-0.18	-0.18
YKL026C	-0.19	-0.15	0.03	0.27	0.54	3.64	2.74
YPR055W	0.15	0.15	0.17	0.09	0.07	0.09	0.07

- If the expression vector of a newly sequenced gene is similar to the expression vector of a gene with known function, a biologist may suspect that these genes perform related functions.
- Genes with similar expression vectors may imply that the genes are co-regulated, meaning that their expression is controlled by the same transcription factor.
- Gene expression analysis is important in biomedical studies such as analyzing tissues before and after a drug is administered or contrasting cancerous and non-cancerous cells

BRAC UNIVERSITY
Inspiring Excellence

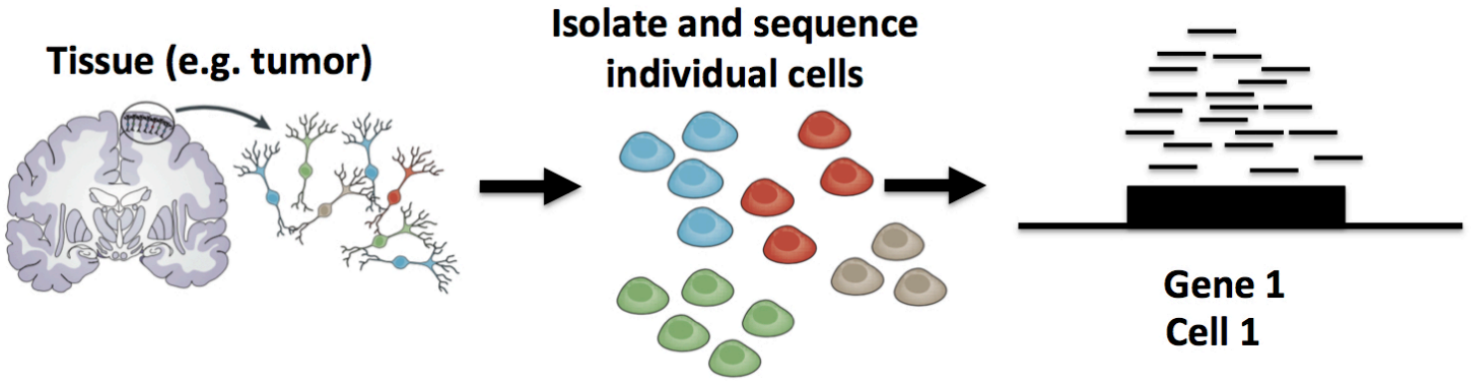
RNA-Seq

- RNA-seq starts by isolating all of the RNA from a cell.
- RT enzyme copies RNA into a set of complementary DNA (cDNA).
- The cDNA is then sequenced using a DNA sequencing machine.



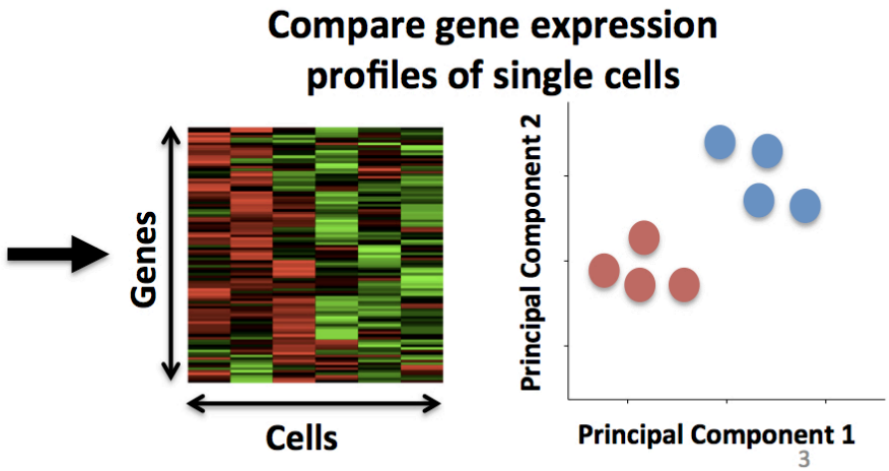
BRAC UNIVERSITY
Inspiring Excellence

Single Cell RNA-Seq



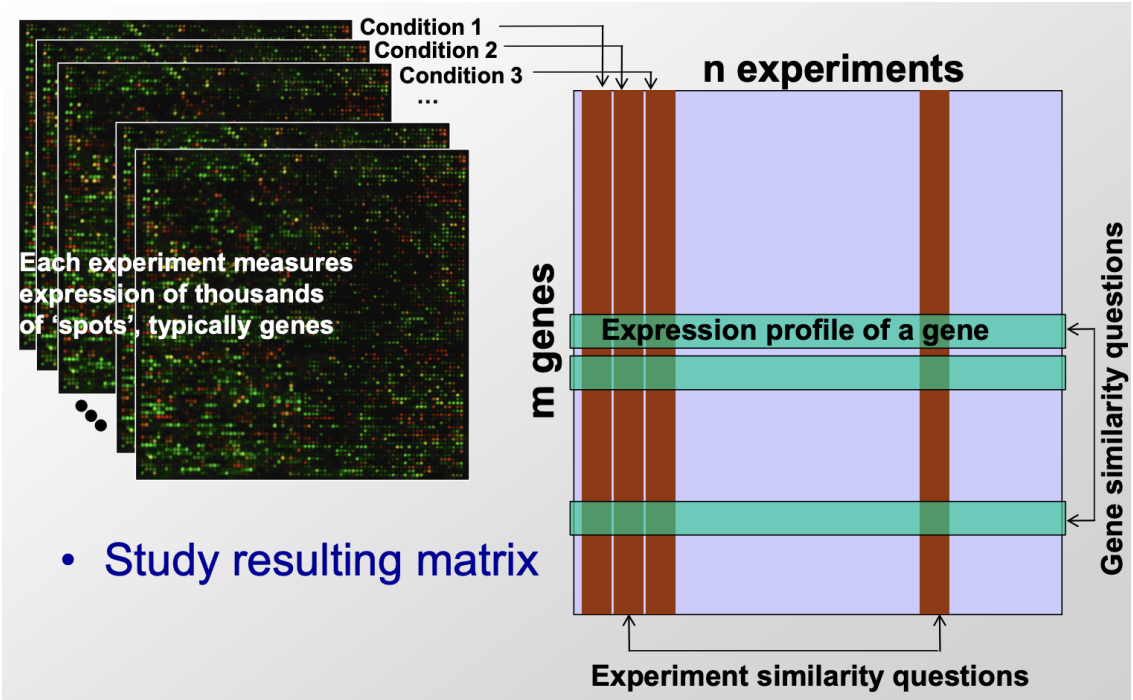
Read Counts

	Cell 1	Cell 2	...
Gene 1	18	0	
Gene 2	1010	506	
Gene 3	0	49	
Gene 4	22	0	
...			



Expression Analysis Data Matrix

- Measure 20,000 genes in 100s of conditions

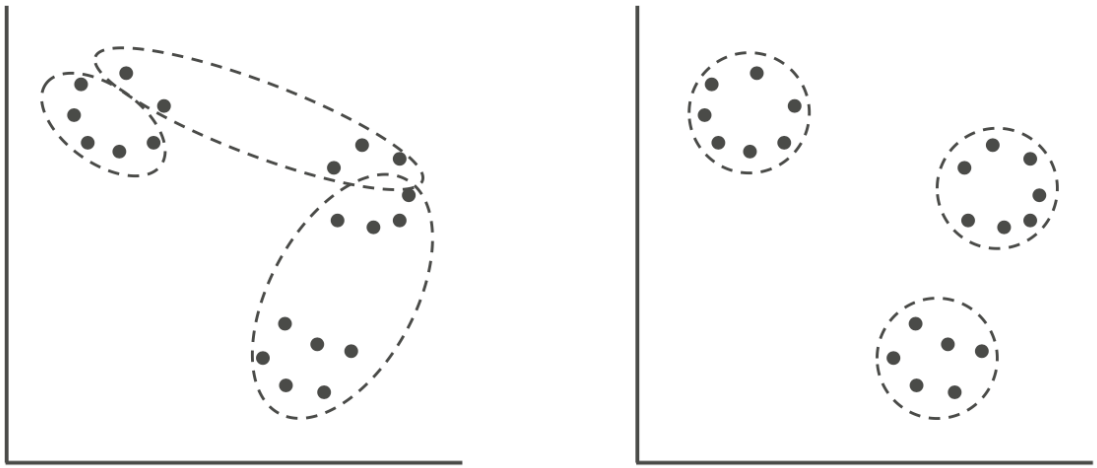


- **Classification:** Extract features from the data that best assign new elements to $\geq$ of well-defined classes
- **Clustering:** Group similar items that likely come from the same category, and in doing so reveal hidden structure.

BRAC UNIVERSITY
Inspiring Excellence

Properties of a Good Clustering

- Every "good" clustering must have two properties:
  - **Homogeneity:** Elements within the same cluster are close to each other.
  - **Separation:** Elements in different clusters are located far from each other.



Clustering - An Optimization Problem

- Rather than thinking about clustering as dividing data points Data into k clusters, we will instead try to select a set Centers of k points that will serve as the centers of these clusters.
- We would like to choose Centers so that they minimize some distance function between Centers and Data over all possible choices of centers.
- Given a point DataPoint v in multi-dimensional space and a set of k points Centers C, we define the distance from DataPoint v to Centers, denoted $d(v, C)$, as the Euclidean distance from DataPoint to its closest center.

$$d(v, C) = \min_{\text{all points } C_i \text{ from C}} d(v, C_i)$$

BRAC UNIVERSITY
Inspiring Excellence

Clustering - An Optimization Problem

- Given a set Data of n data points and a set Centers of k centers, the squared error distortion of Data and Centers, denoted $distortion(Data, C)$, is defined as the mean squared distance from each data point to its nearest center.

$$distortion(Data, C) = \frac{1}{n} \sum_{\text{all points } p \text{ in Data}} d(p, C)^2$$

k-Means Clustering Problem

Given a set of data points, find k center points minimizing the squared error distortion.

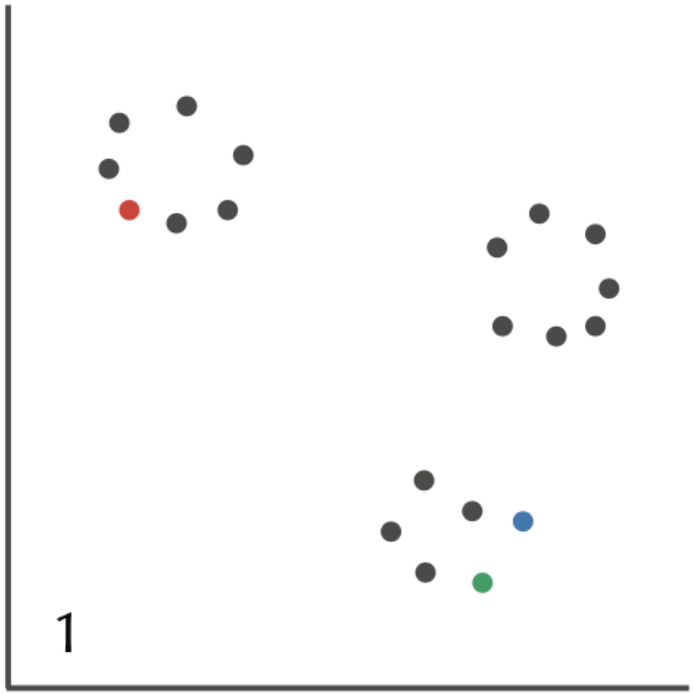
- **Input:** A set of points Data and an integer k.
- **Output:** A set Centers of k centers that minimize distortion(Data, C) over all possible choices of k centers.

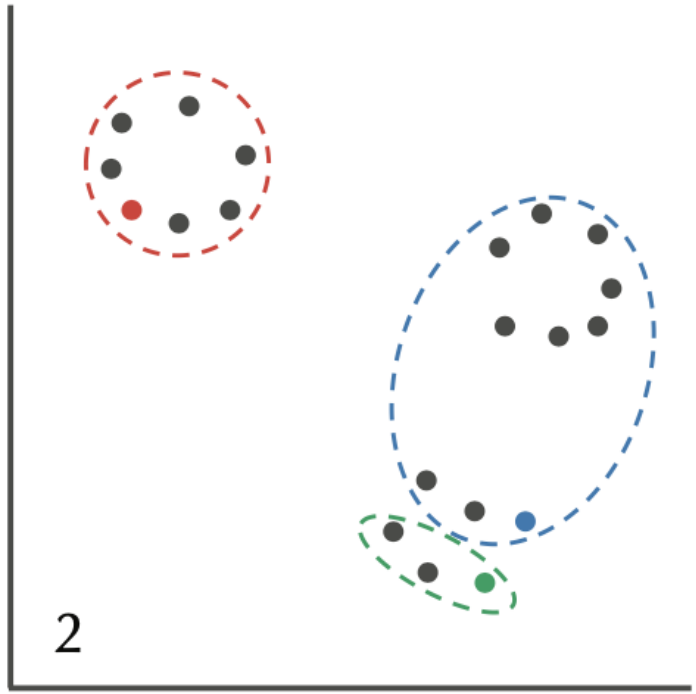
The Lloyd Algorithm

1. Arbitrarily assign the k cluster centers
2. while the cluster centers keep changing
   3. **Centers to Clusters:** Assign each data point to the cluster C_i corresponding to the closest cluster representative (center) $(1 \leq i \leq k)$
   4. **Clusters to Centers:** After the assignment of all data points, compute new cluster representatives according to the center of gravity of each cluster

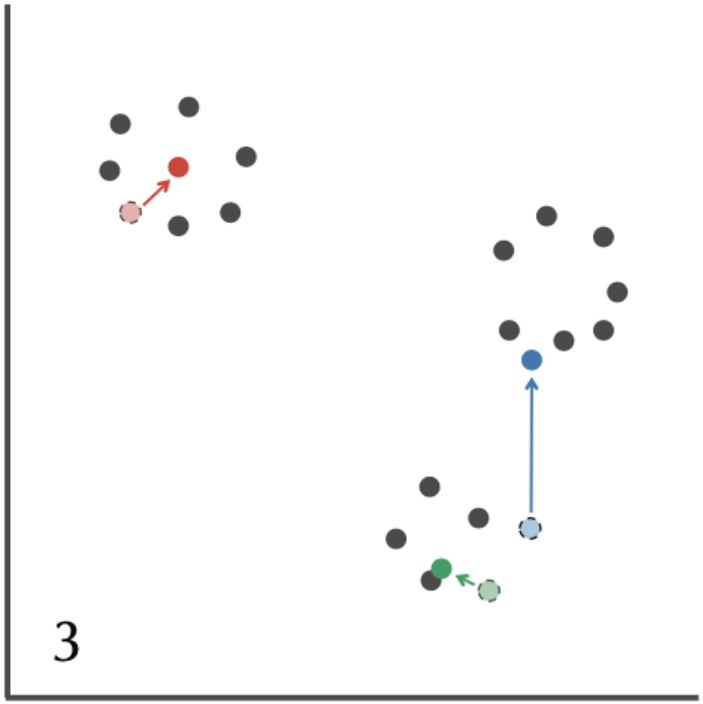
- May stuck into a local minima
- Depends on the initialization
- Selection of k
- Assumption of spherical clusters
- Poor handling of outliers and noise
- High computational cost
- Fails for higher dimensional data
- No categorical data

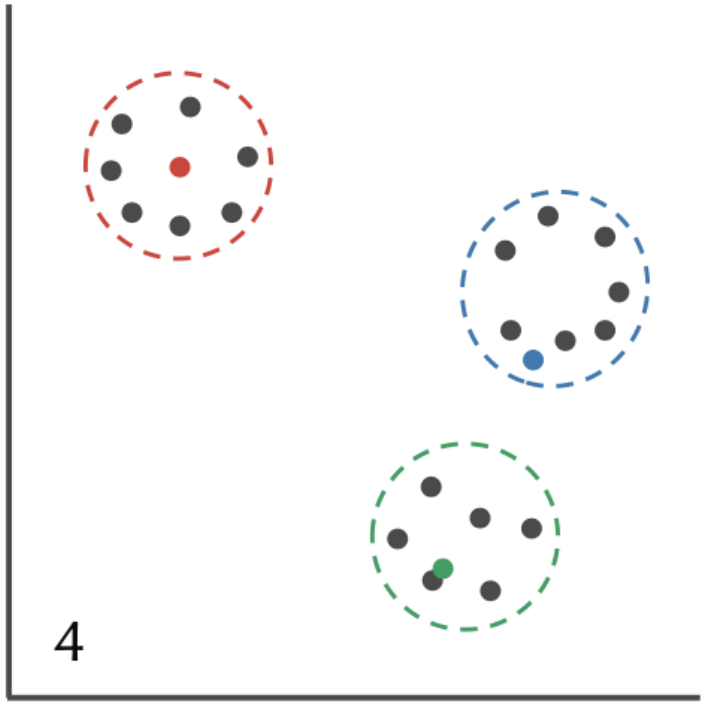
The Lloyd Algorithm



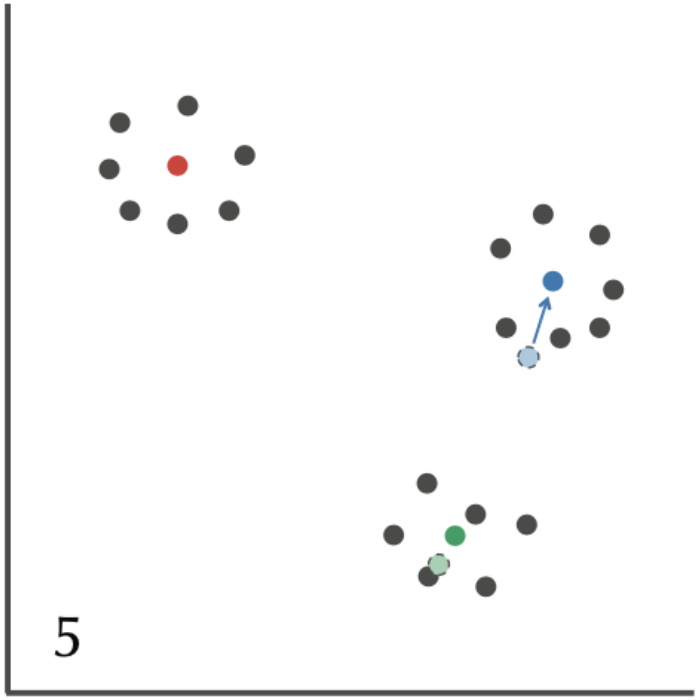


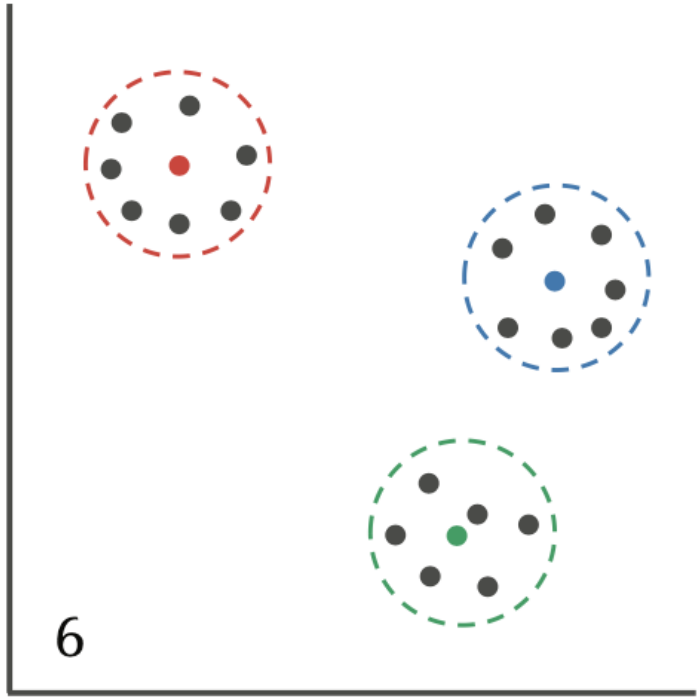
The Lloyd Algorithm





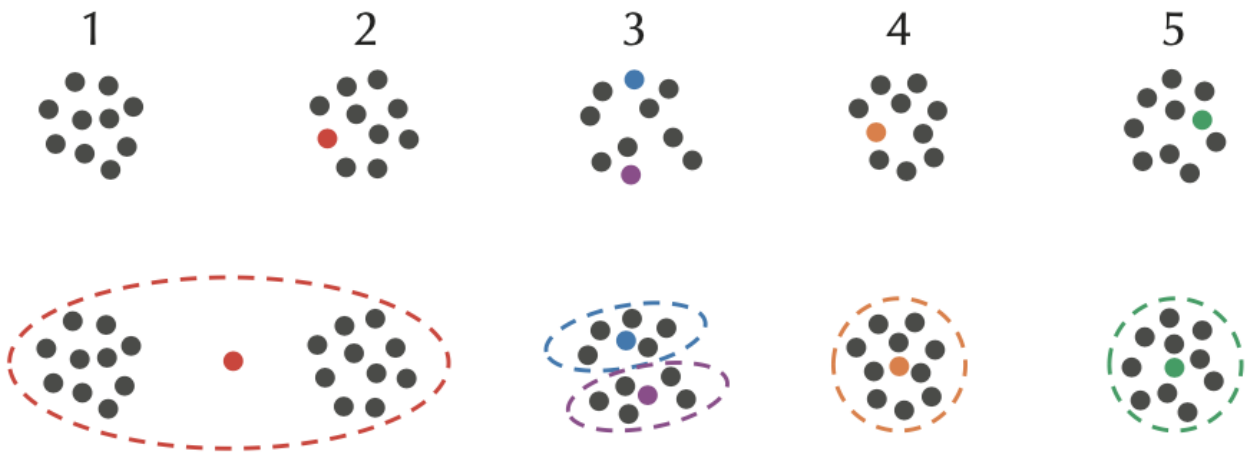
The Lloyd Algorithm





BRAC UNIVERSITY
Inspiring Excellence

Initialization in KMeans!



***k*-MEANS++ INITIALIZER**(*Data*, *k*)

Centers ← the set consisting of a single randomly chosen point from *Data*

while |*Centers*| < *k*

randomly select *DataPoint* from *Data* with probability proportional to $d(DataPoint, Centers)^2$

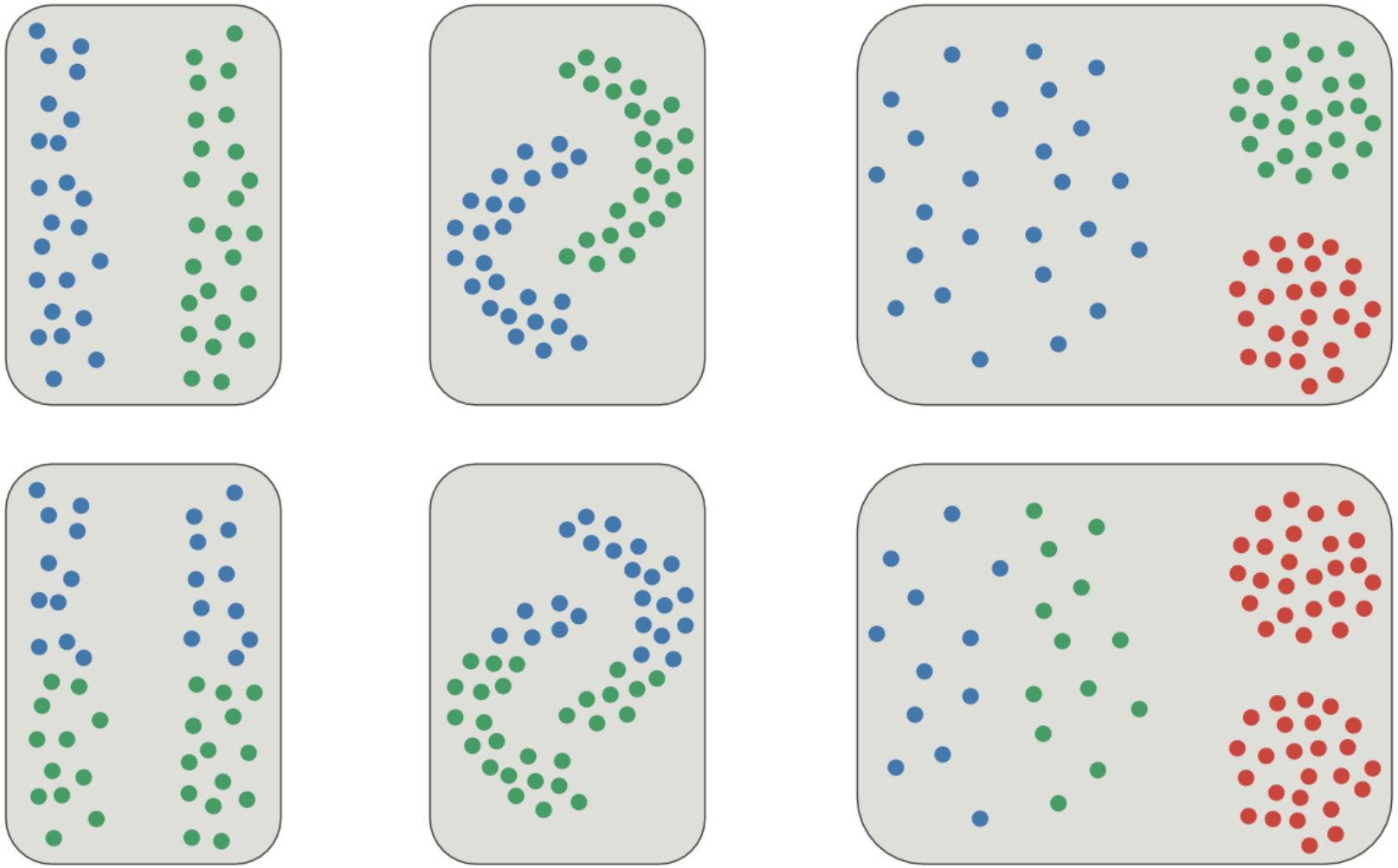
add *DataPoint* to *Centers*

return *Centers*

BRAC UNIVERSITY
Inspiring Excellence

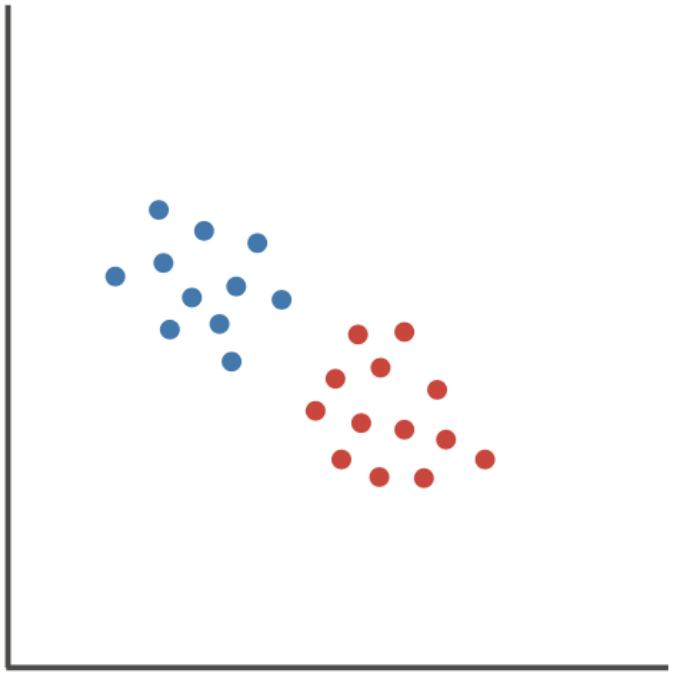
Limitations of Kmeans

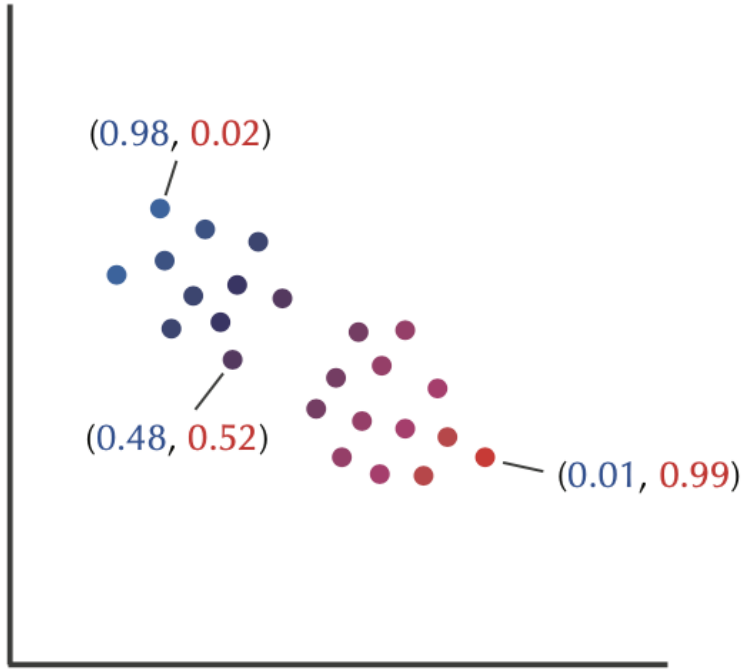
- Assumption of spherical clusters



Limitations of Kmeans

- *k*-Means clustering forces us to make a "hard" assignment of each point to only one cluster.
- This strategy makes little sense for midpoints, or points that are approximately equidistant from two centers.





EM Algorithm - Soft Kmeans Clustering

1. This algorithm starts from randomly chosen centers
2. Iterates the following two steps:
   3. **Centers to Soft Clusters (E-step):** After centers have been selected, assign each data point a "responsibility" for each cluster, where higher responsibilities corre- spond to stronger cluster membership.
   4. **Soft Clusters to Centers (M-step):** After data points have been assigned to soft clusters, compute new centers.

Given k centers Centers $= (x_1, \cdots x_k)$ and n points $Data = (Data_1, \cdots, Data_n)$, we therefore need to construct a $k \times n$ responsibility matrix HiddenMatrix for which $HiddenMatrix_{i,j}$ is the pull of center i on data point j.

EM Algorithm

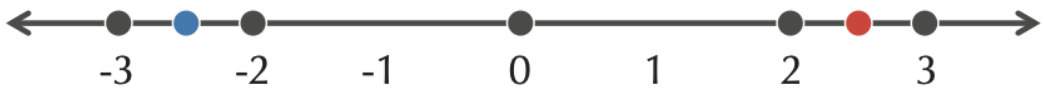
$$HiddenMatrix_{i,j} = \frac{1/d(Data_j, x_i)^2}{\sum_{\text{all centers } x_i} 1/d(Data_j, x_i)^2}.$$

$$HiddenMatrix_{i,j} = \frac{e^{-\beta \cdot d(Data_j, x_i)}}{\sum_{\text{all centers } x_i} e^{-\beta \cdot d(Data_j, x_i)}}$$

- In the M step, updated center x_i is a weighted center of gravity of the points Data.

BRAC UNIVERSITY
Inspiring Excellence

EM Algorithm



0.992	0.988	0.500	0.012	0.008
0.008	0.012	0.500	0.988	0.992
0.924	0.881	0.500	0.119	0.076
0.076	0.119	0.500	0.881	0.924
0.993	0.982	0.500	0.118	0.007
0.007	0.018	0.500	0.982	0.993

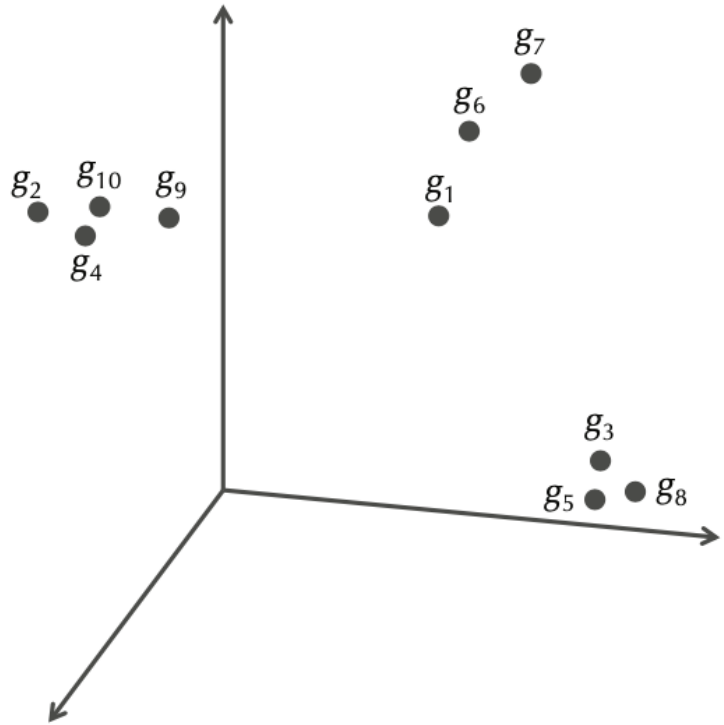
FIGURE 8.21 (Top) Five one-dimensional points *Data* = (-3, -2, 0, +2, +3) with two centers (shown in blue and red) *Centers* = {-2.5, +2.5}. (Bottom) Three versions of *HiddenMatrix* constructed for *Data* and *Centers*, using the Newtonian inverse-square law (first matrix) and the partition function with stiffness $\beta = 0.5$ (second matrix), and $\beta = 1$ (third matrix).

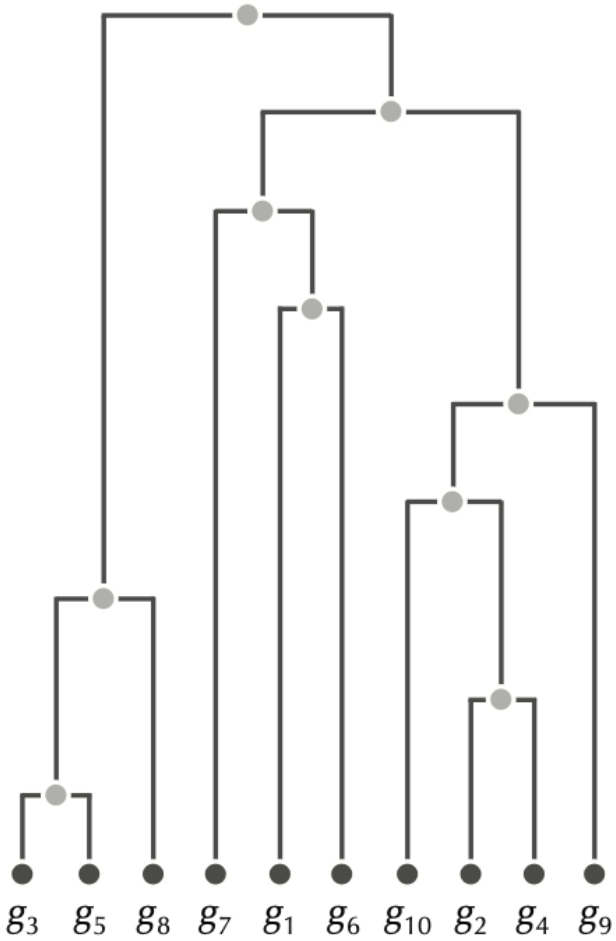
$$x_1 = \frac{0.993 \cdot (-3) + 0.982 \cdot (-2) + 0.500 \cdot (0) + 0.018 \cdot (2) + 0.007 \cdot (3)}{0.993 + 0.982 + 0.500 + 0.018 + 0.007} = -1.955$$

$$x_2 = \frac{0.007 \cdot (-3) + 0.018 \cdot (-2) + 0.500 \cdot (0) + 0.982 \cdot (2) + 0.993 \cdot (3)}{0.007 + 0.018 + 0.500 + 0.982 + 0.993} = 1.955$$

BRAC UNIVERSITY
Inspiring Excellence

Hierarchical Clustering





BRAC UNIVERSITY
Inspiring Excellence

Hierarchical Clustering

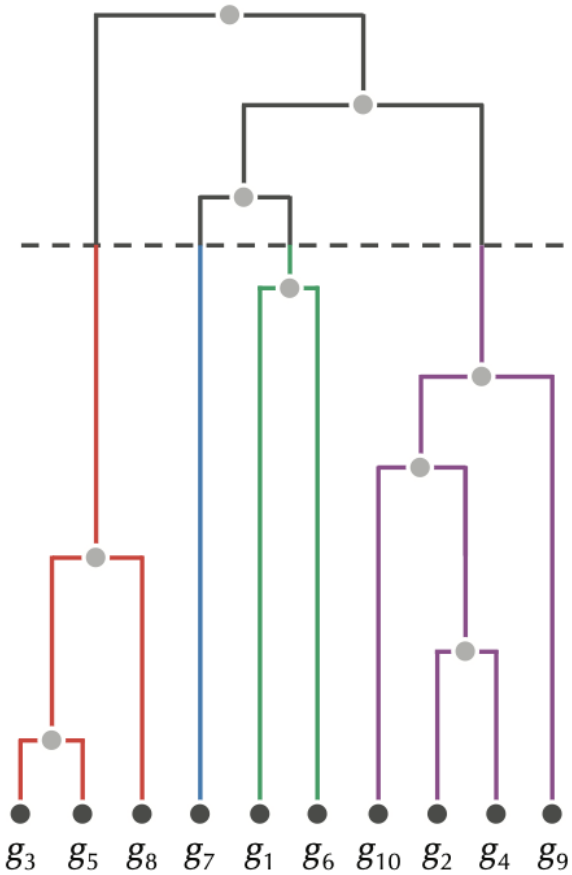
- Plot each datum as a point in N-dimensional space.
- Create a distance matrix for the distance between every two gene points in the N-dimensional space.
- Genes with a small distance share the same expression characteristics and might be functionally related or similar.
- Therefore, clustering "close" genes together reveals groups of functionally related genes

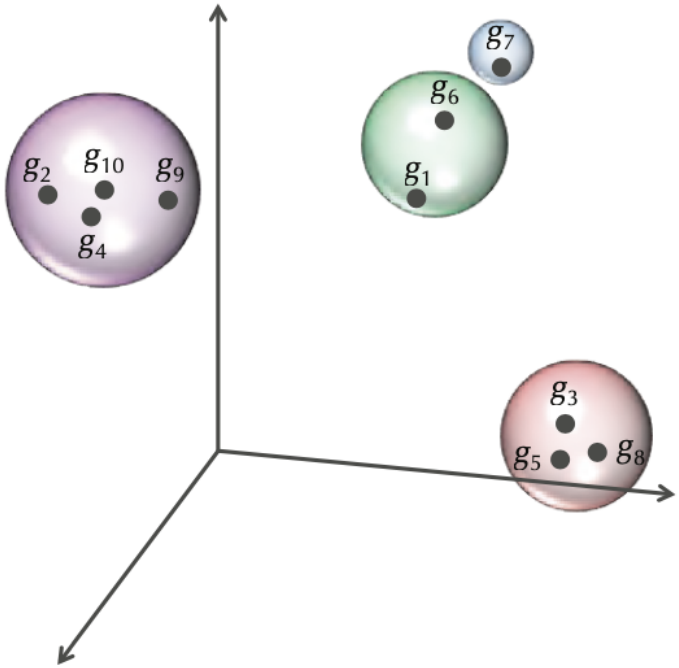
	1 hr	2 hr	3 hr
g_1	10.0	8.0	10.0
g_2	10.0	0.0	9.0
g_3	4.0	8.5	3.0
g_4	9.5	0.5	8.5
g_5	4.5	8.5	2.5
g_6	10.5	9.0	12.0
g_7	5.0	8.5	11.0
g_8	3.7	8.7	2.0
g_9	9.7	2.0	9.0
g_{10}	10.2	1.0	9.2

	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

BRAC UNIVERSITY
Inspiring Excellence

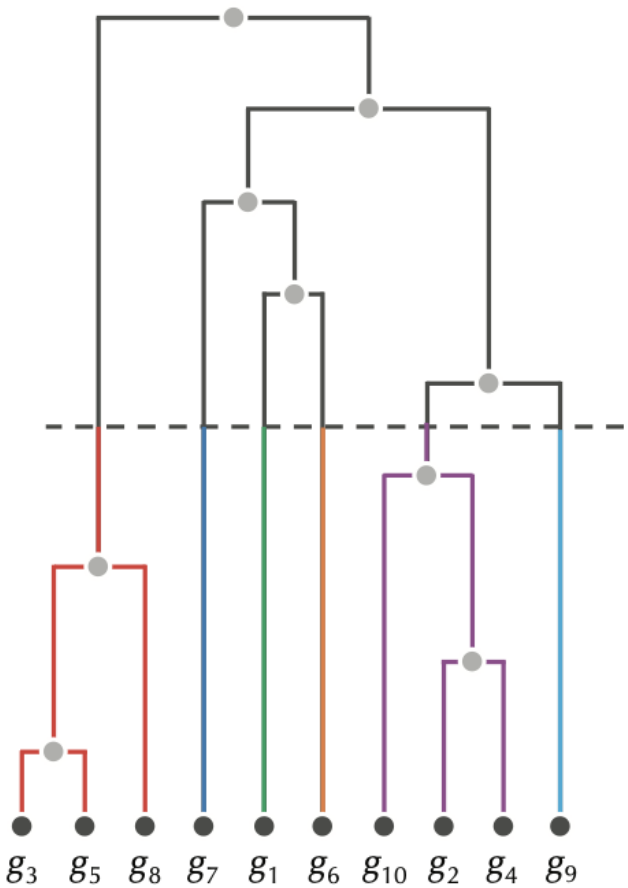
Hierarchical Clustering Cuts

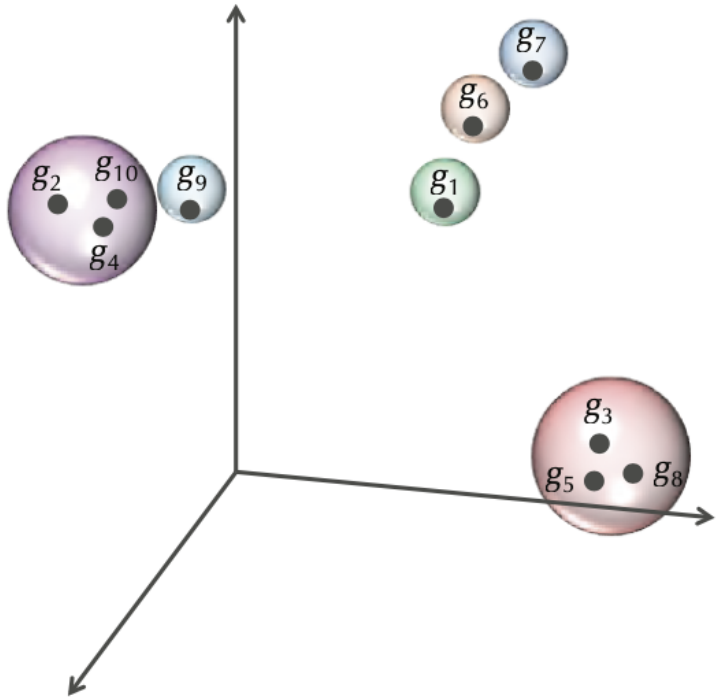




BRAC UNIVERSITY
Inspiring Excellence

Hierarchical Clustering Cuts





BRAC UNIVERSITY
Inspiring Excellence

Hierarchical Clustering Algorithm

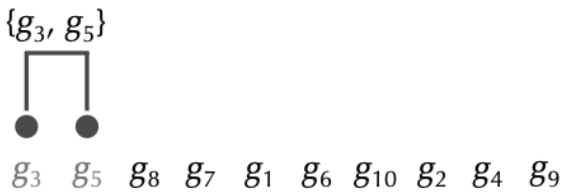
HierarchicalClustering(D, n)

Clusters $\leftarrow$ n single-element clusters labeled $1, \ldots, n$
construct a graph T with n isolated nodes labeled by single elements $1, \ldots, n$
while there is more than one cluster
  find the two closest clusters C_i and C_j (break ties arbitrarily)
  merge C_i and C_j into a new cluster C_{new} with $|C_i| + |C_j|$ elements
  add a new node labeled by cluster C_{new} to T
  connect node C_{new} to C_i and C_j by directed edges
  remove the rows and columns of D corresponding to C_i and C_j
  remove C_i and C_j from *Clusters*
  add a row/column to D for C_{new} by computing $D(C_{new}, C)$ for each C in *Clusters*
  add C_{new} to *Clusters*
root $\leftarrow$ the node in T corresponding to the remaining cluster
return T

BRAC UNIVERSITY
Inspiring Excellence

Hierarchical Clustering Algorithm

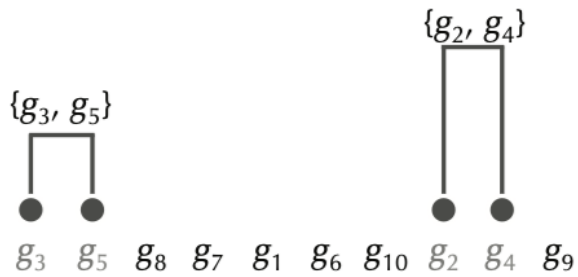
	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	**0.7**	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1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



BRAC UNIVERSITY
Inspiring Excellence

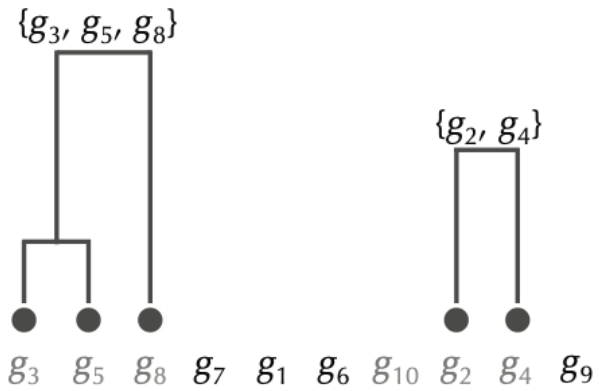
Hierarchical Clustering Algorithm

	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



Hierarchical Clustering Algorithm

	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



BRAC UNIVERSITY
Inspiring Excellence

Hierarchical Clustering Algorithm

- Distance $D(C_{new}, C)$ between a newly formed cluster C_{new} and each old cluster C.

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

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