STAT540
Lecture 16: March 7th 2016

Clustering: problem, objectives, and algorithms

Sara Mostafavi
Department of Statistics
Department of Medical Genetics
Center for Molecular Medicine and Therapeutics

** Many thanks to Drs. Gabriela Cohen-Freue and Jenny Bryan for lecture slides**

announcement

- Paper review 1
- Project's progress report

- Homework 1 posted - March 18th
- Paper review 2 - April 1st

- Seminars!!

January	4	1	Intro to course	Paul
January	6	2	Review of probability & inference	Sara
January	11	3	Review of probability & inference	Sara
January	13	4	Exploratory data analysis	Paul
January	18	5	Data QC and preprocessing	Sara
January	20	6	Statistical inferene: two group comparisons	Sara
January	25	7	Statistical inferene: more than two groups	Sara
January	27	8	Statistical inference: linear models	Sara
February	1	9	Statistical inference: linear models	Sara
February	3	10	Statistical inference: large scale, empirical bayes	Sara
February	10	11	Statistical inference: multiple testing	Bernard
February	22	12	RNA-Seq data analysis I	Paul
February	24	13	RNA-Seq data analysis II	Paul
February	29	14	Epigenetic data	Meaghan
March	2	15	PCA	Paul
March	8	16	Clustering	Sara
March	9	17	Classification	Sara
March	14	18	Cross validation	Sara
March	16	19	Regularization	Sara
March	21	20	Permuatation analysis, bootstrap	Sara
March	23	21	Analysis of gene function I	Paul
March	30	22	Analysis of gene function II	Paul
April	4	23	Guest lecture: Sohrab Shah	GL
April	6	24	Poster session	

Statistical Essentials

"Real-World" Data

High Dimensional Data Analysis (ML)

Interpreting the "Findings"

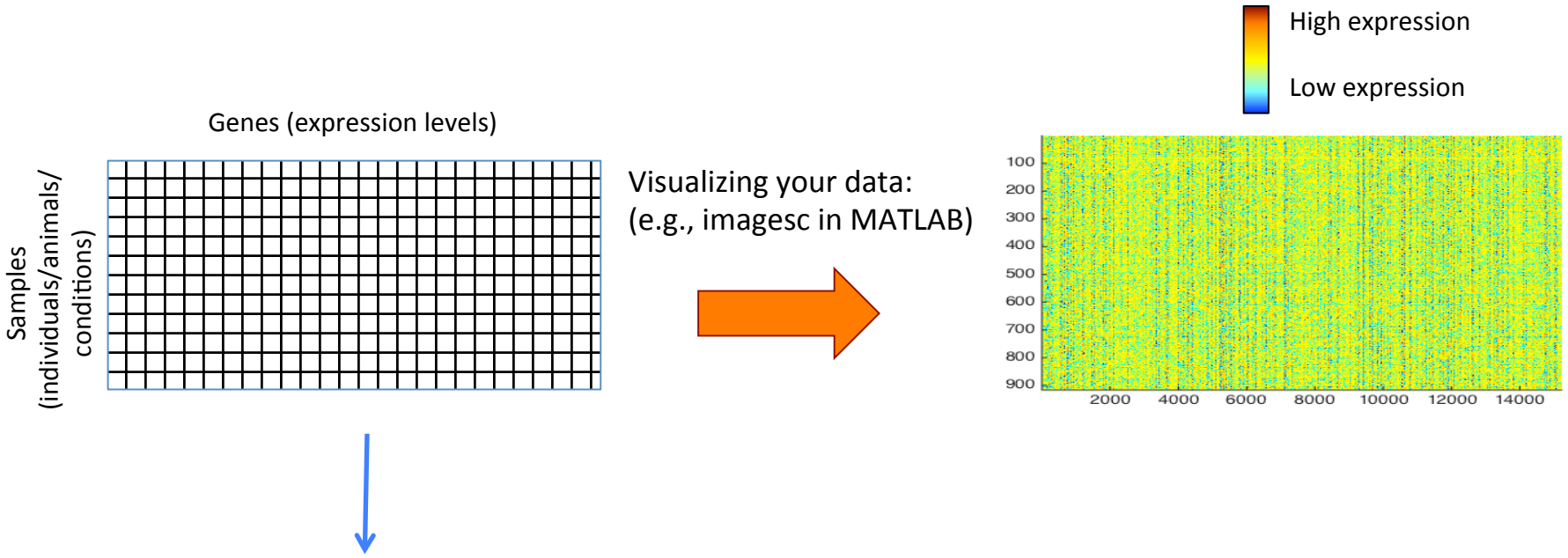
How is machine learning different from statistical inference?

Machine learning is a sub-field of computer science:

- Roots go back to Artificial Intelligence research (60-80s)
- Incorporated and built upon statistical inference
- Cultural/philosophical differences led to special insights
- If you don't have a good grasp of ½ half of the course, ML approaches becomes black-boxes

- The biggest difference: statistics emphasizes inference, ML emphasized prediction. Inferring the **process/mechanism** by which data was generated vs predicting what future data will look like given current data/metadata. (Flipping of X and Y!)
- Because of this, ML has focused on "high dimensional" problems from infancy --- high dimensionality is desirable: more data better predictions?

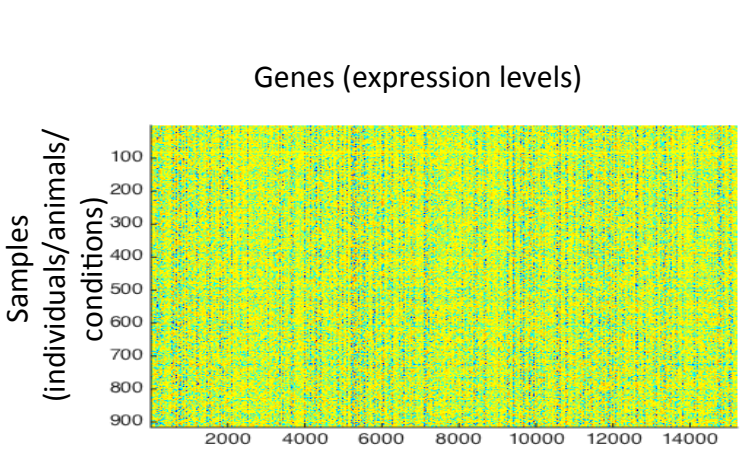
Visualizing “raw” expression data (without clustering) is NOT informative…



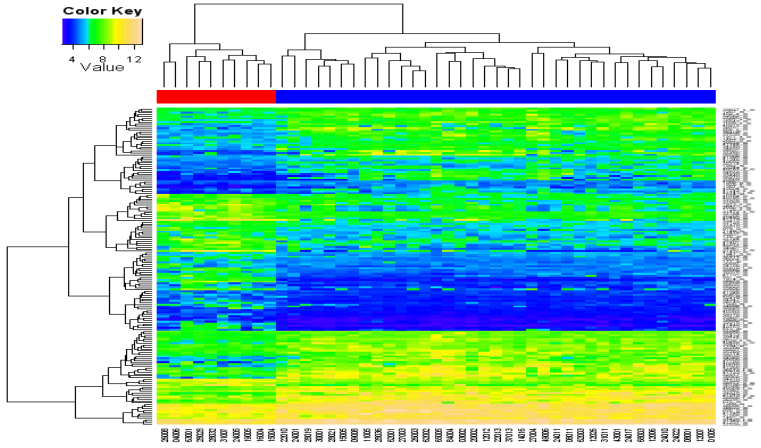
Matrix X with dimensions n by p (n rows and p columns)

Pervasive application of clustering in analysis of gene expression data

A more familiar picture seen in “omics” papers ….

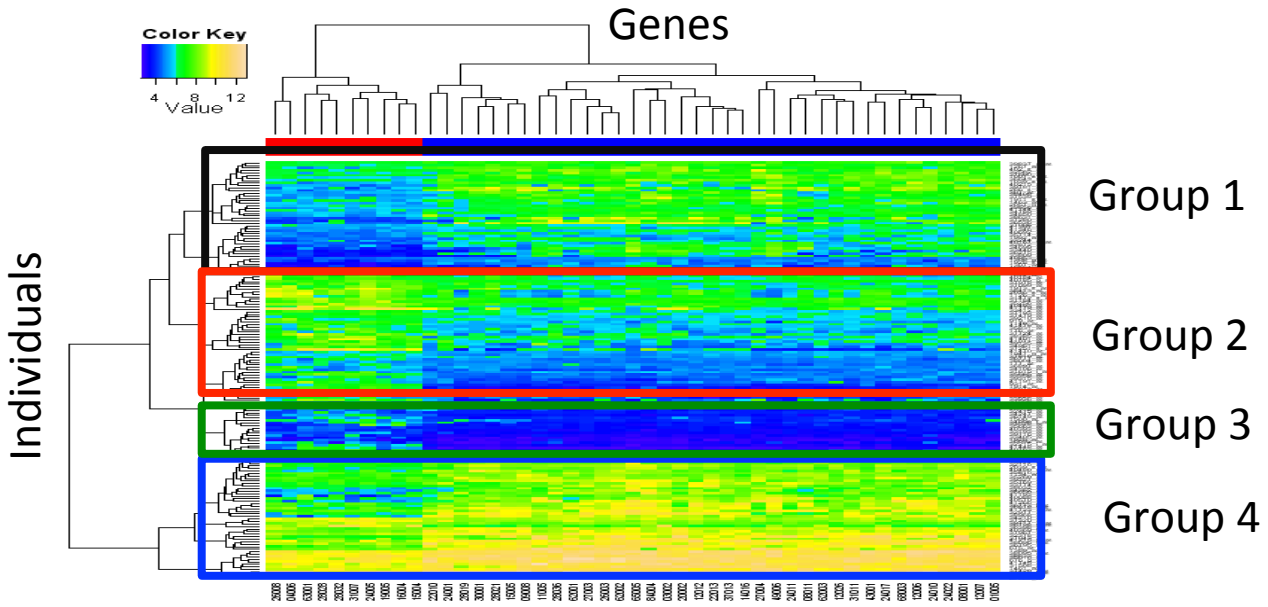


Clustering algorithm



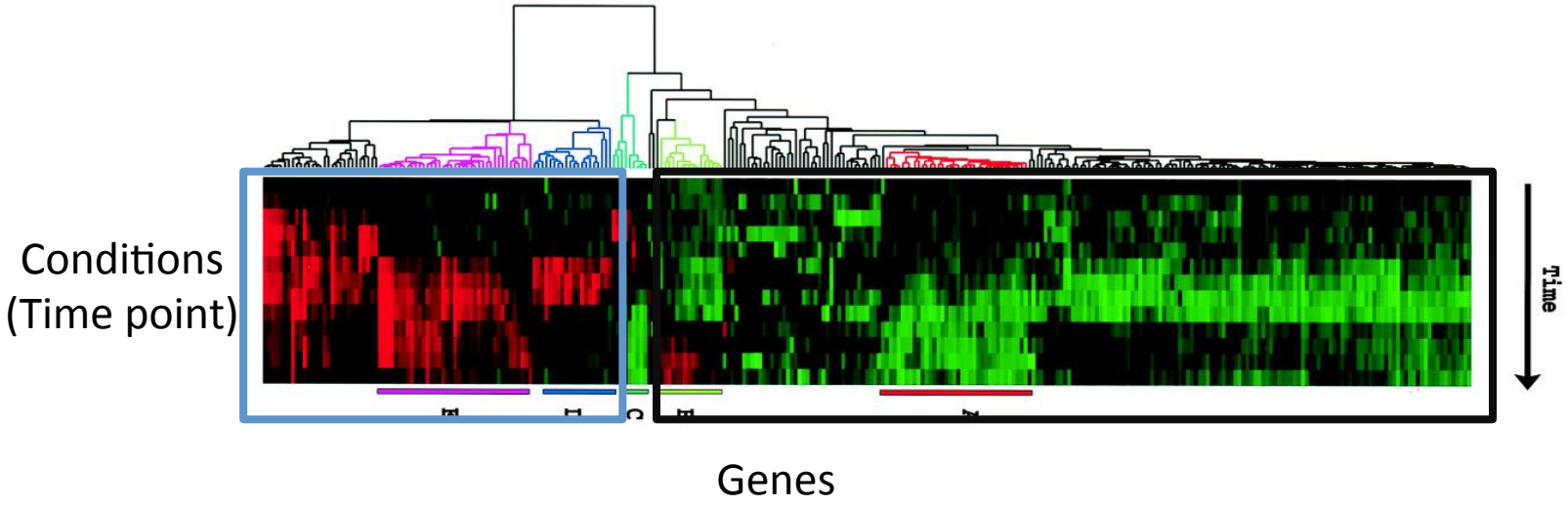
Two predominant application of clustering in gene expression studies

1. Identify groups of individuals that have similar expression profiles:
   - Identifying disease sub-types



Two predominant application of clustering in gene expression studies

2. Identify groups of genes that have correlated expression profiles:
   - Informative of co-functionality (genes in the same “cluster” perform the same function)



What is Clustering?

- "Clustering" Colloquially means placing/grouping a set of objects into groups/clusters.
- Clustering is a formal **problem** in Computer Science and in Statistics, with formal definitions and "solutions".
- Clustering in bioinformatics is often used as a tool for visualization, hypothesis generation, selection of genes for further analysis.
  - Keep in mind, with typical use of clustering in bioinformatics: there is no measure of "strength of evidence" or "strength of clustering structure" provided.
- Rigorous application of clustering is very powerful but also hard to do (computational complexity, suitable definition of clustering objective, determining the number of clusters …)

Clustering problem: definition

- Goal: place a set of **objects** into groups or **clusters** in a way that **similar** objects are in the same cluster.

Cluster some rocks:

Clustering problem: definition

- Goal: place a set of **objects** into groups or **clusters** in a way that **similar** objects are in the same cluster.

Cluster some rocks:

Rocks were clustered according to their color and texture.

Clustering problem: definition

- Goal: place a set of **objects** into groups or **clusters** in a way that **similar** objects are in the same cluster.

Cluster some rocks:

Note that you could have also considered a 2-cluster solution.

Clustering: definitions

- Goal: place a set of objects into groups or **clusters**.
- How do we do this?
  - gather a set of attributes for each object.
  - Place objects in clusters so that objects within each cluster are more similar to each other compared to objects that outside their group/cluster.

Rocks were clustered according to their color and texture.

A clustering objective function

- Goal (the clustering problem):place a set of **objects** into groups or **clusters** in a way that **similar** objects are in the same cluster.
- How do we do this?
  - gather a set of attributes for each object.
  - Place objects in clusters so that objects within each cluster are more similar to each other, based on their attributes, compared to objects that outside their group/cluster.

→ Clustering **objective** function: maximize within cluster similarity

- A precise definition of "good/optimal" clustering: precise enough to be translated into an equation.

Defining attribute/feature vector for each object

- We need to numerically define a attribute or feature vector that describes the relevant properties of each object

Set of n objects $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \ldots, \vec{x}_n\}$

Each object is represented by a numerical vector: $\vec{x}_i$

Attribute/feature p for object i

Rock i: $\vec{x}_i = (x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,p})$

Numerical value representing texture

Numerical value for color/shade

Commonly Used Measures of Similarity and Distance

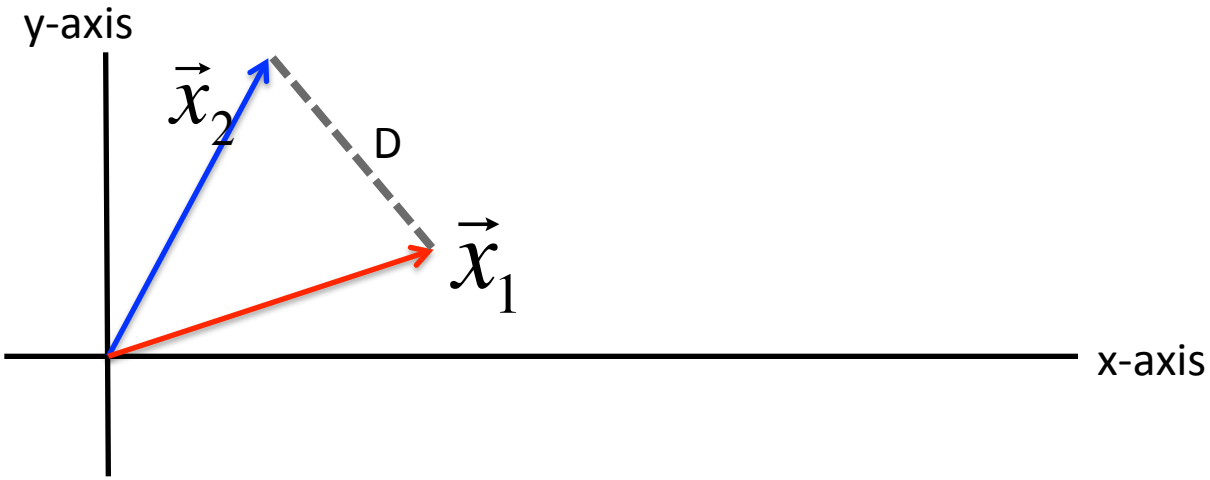
- Every clustering method is based on the measure of distance or similarity.
- We need to compute pairwise similarities between all objects.

- Typical distance/similarity measures:
  - Distance:
    - Euclidean
    - Manhattan
  - Similarity: Correlation
    - Spearman
    - Pearson

Commonly used measures of similarity and distance

- Euclidian distance between two feature vectors: $\vec{x}_1$ and $\vec{x}_2$

$$D = \| \vec{x}_1 - \vec{x}_2 \|_2 = \sqrt{\sum_{j=1}^{p} (x_{1,j} - x_{2,j})^2}$$



Commonly used measures of similarity and distance

- 1-Pearson correlation coefficient: $\vec{x}_1$ and $\vec{x}_2$

$$r = \frac{(\vec{x}_1 - \mu_1) \bullet (\vec{x}_2 - \mu_2)}{\sigma_x \sigma_y}$$

If you standardize your variables/features

$$r = \vec{x}_1 \bullet \vec{x}_2 = \sum_{j=1}^{p} x_{1,j} \times x_{2,j}$$

$$D^2 = \sum_{j=1}^{p} (x_{1,j} - x_{2,j})^2 = \sum_{j=1}^{p} (x_{1,j}^2 + x_{2,j}^2) - 2\sum_{j=1}^{p} (x_{1,j}^2 \times x_{2,j}^2)$$

Commonly used measures of similarity and distance

- 1-Pearson correlation coefficient: $\vec{x}_1$ and $\vec{x}_2$

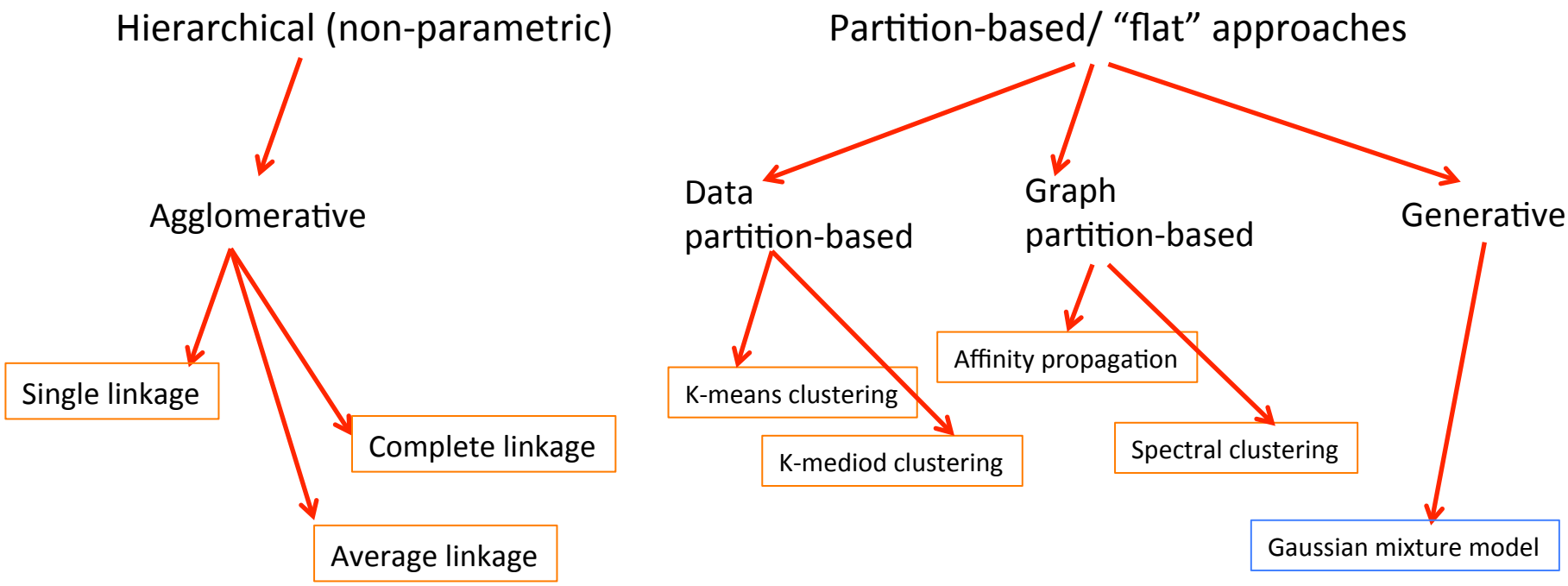
$$r = \frac{(\vec{x}_1 - \mu_1) \bullet (\vec{x}_2 - \mu_2)}{\sigma_x \sigma_y}$$

If you standardize your variables/features

$$r = \vec{x}_1 \bullet \vec{x}_2 = \sum_{j=1}^{p} x_{1,j} \times x_{2,j}$$

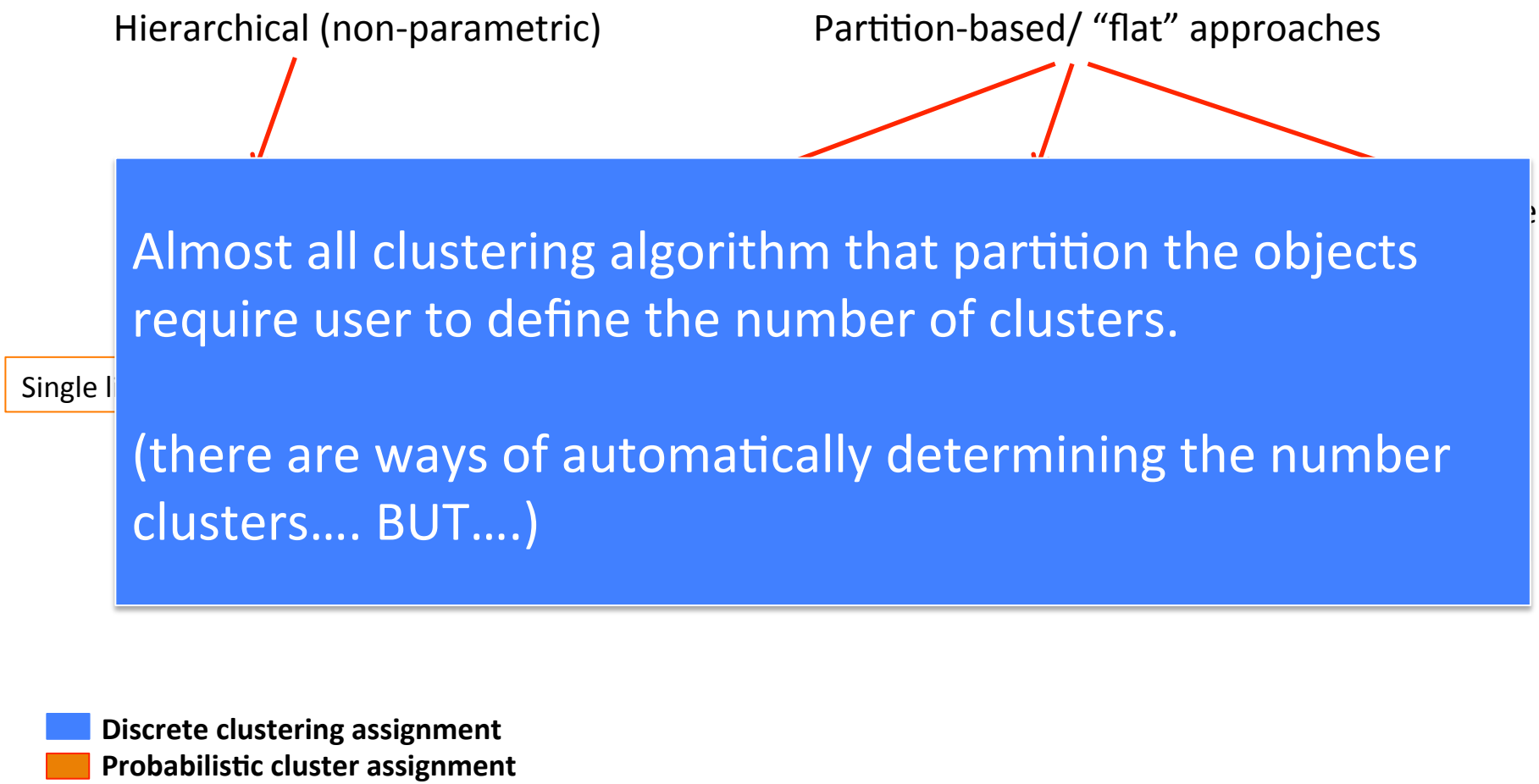
$$D^2 = \sum_{j=1}^{p} (x_{1,j} - x_{2,j})^2 = \sum_{j=1}^{p} (x_{1,j}^2 + x_{2,j}^2) - 2\sum_{j=1}^{p} (x_{1,j}^2 \times x_{2,j}^2) = 2(1 - r)$$

Some existing clustering approaches



Discrete clustering assignment
Probabilistic cluster assignment

Some existing clustering algorithms



Three key concepts with distinct definitions:

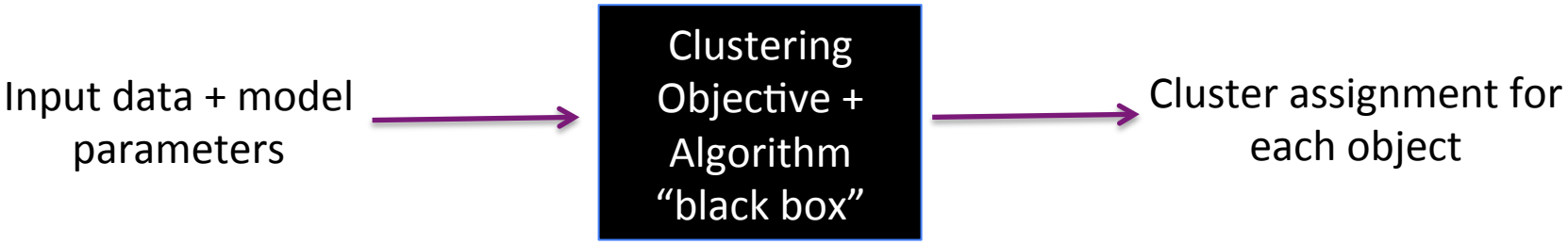
1) The clustering problem
2) A clustering objective function (model)
3) A clustering algorithm

What is an algorithm?

- An algorithm is a self-contained step-by-step set of operations to be performed in order to achieve a given task. Through a set of steps, an algorithm transforms a given input data into the desired output.

(from wikipedia)

Clustering algorithm from a machine learning perspective: what are the inputs and the outputs?



Input: 1) data matrix X_{nxp} (rows are the objects)

2) number of clusters k

Output: an assignment of cluster membership for each object. $C_{nx1} = \{1,k\}^n$, C_i=k if object i is placed in cluster k.

(Note the output vector can also be a probabilistic assignment, we'll ignore this for now.)

K-means clustering objective function

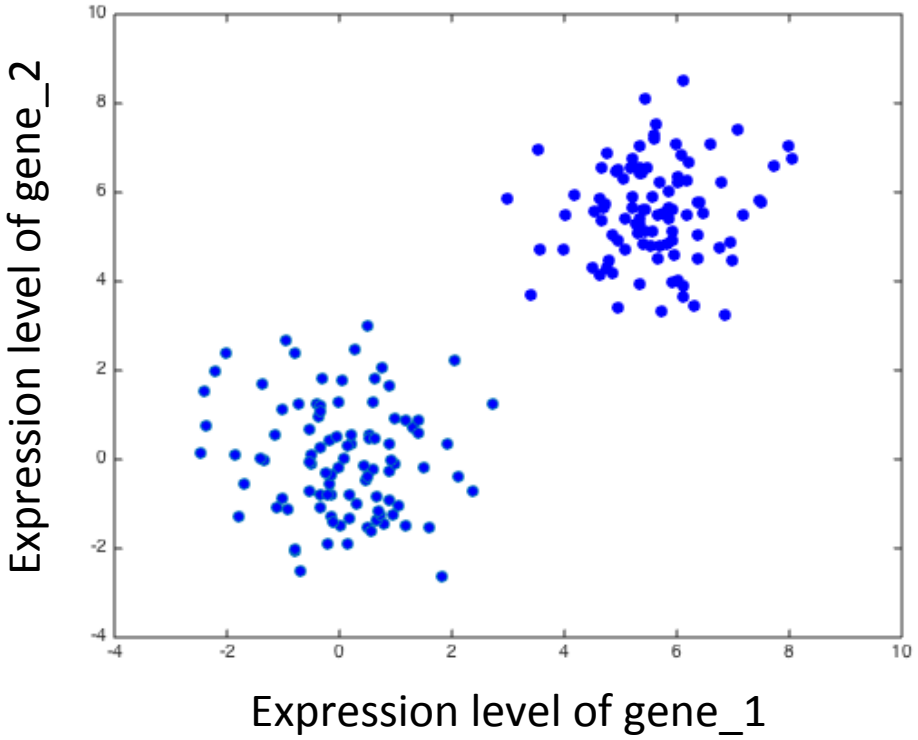
- One of the most widely used partition-based clustering approaches.
- **Objective function**: minimize the average squared Euclidean distance of objects from their assigned cluster centers. A cluster center (or centroid) is defined as the mean of objects in the given cluster.

K-means clustering objective function

- One of the most widely used partition-based clustering approaches.
- **Objective function**: minimize the average squared **Euclidean distance** of objects from their assigned cluster centers. A **cluster center** (or centroid) is defined as the mean of objects in the given cluster.

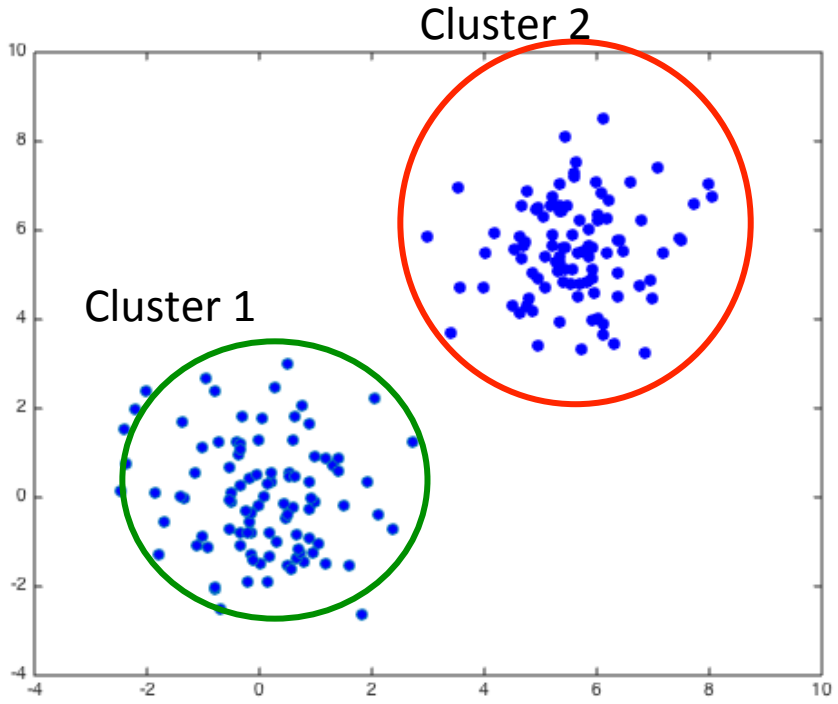
How many clusters are there?

Suppose you measured expression levels for 2 genes (gene A and gene B) for 200 individuals



How many clusters are there?

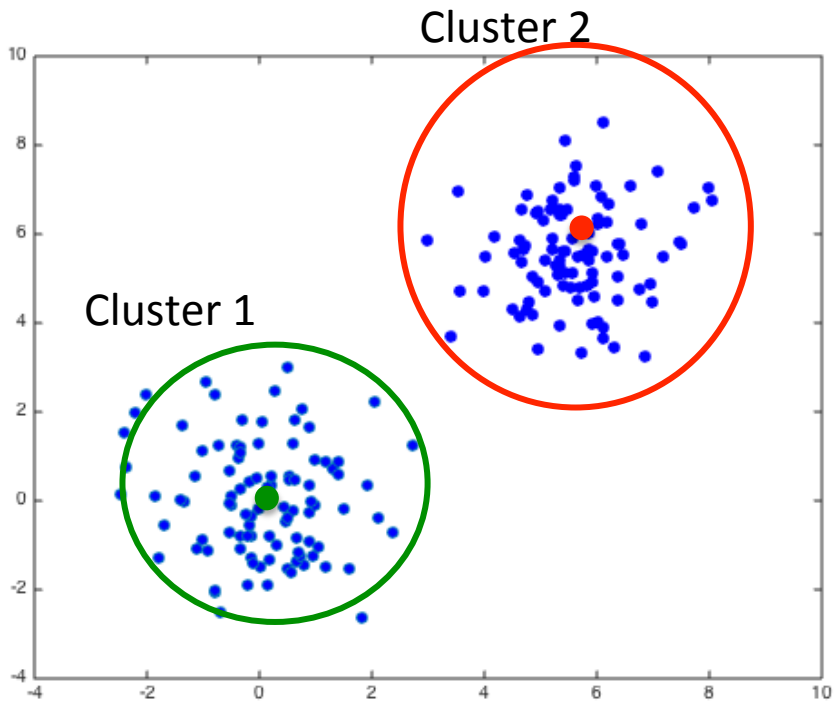
Suppose you measured expression levels for 2 genes (gene A and gene B) for 100 individuals



K-means objective function

Objective function: minimize the average squared **Euclidean distance** of objects from their assigned cluster centers. A **cluster center** (or centroid) is defined as the mean of objects in the given cluster.

Computing the "mean/centroid" for each cluster:

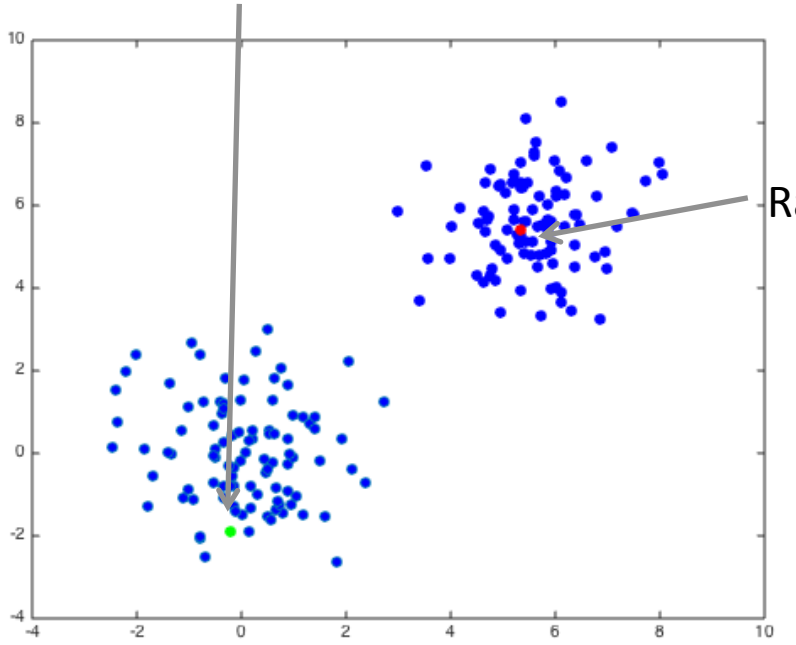


Let's go through the k-means algorithm first

Algorithm: iterative procedure

1) Pick k random points as initial cluster centers

Randomly selected point 1



Randomly selected point 2

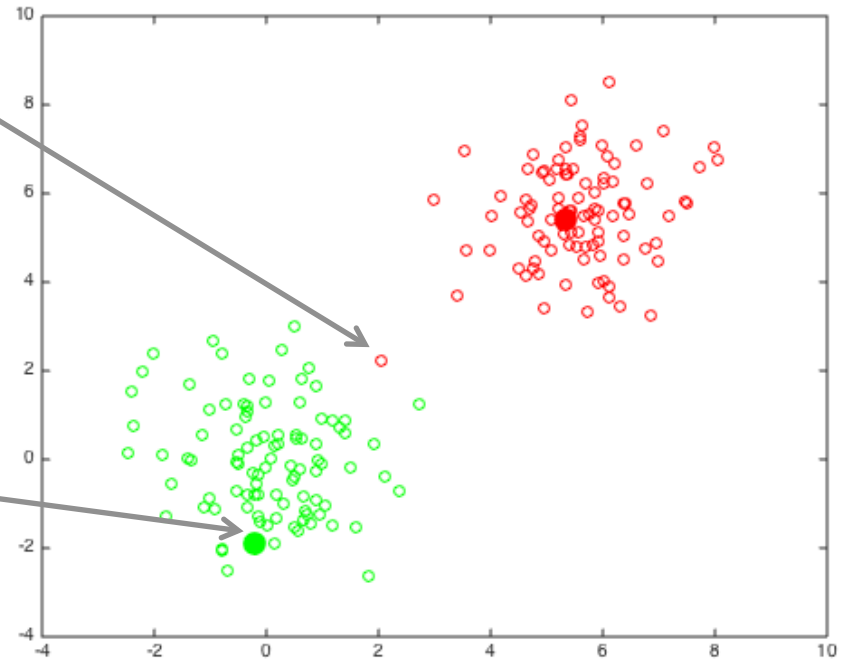
Let's go through the k-means algorithm first

Algorithm: iterative procedure

1) Pick k random points as initial cluster centers
2) Measure distance between all points and the cluster centers – assign points to nearest cluster

Doesn't look right

"Initialization point"



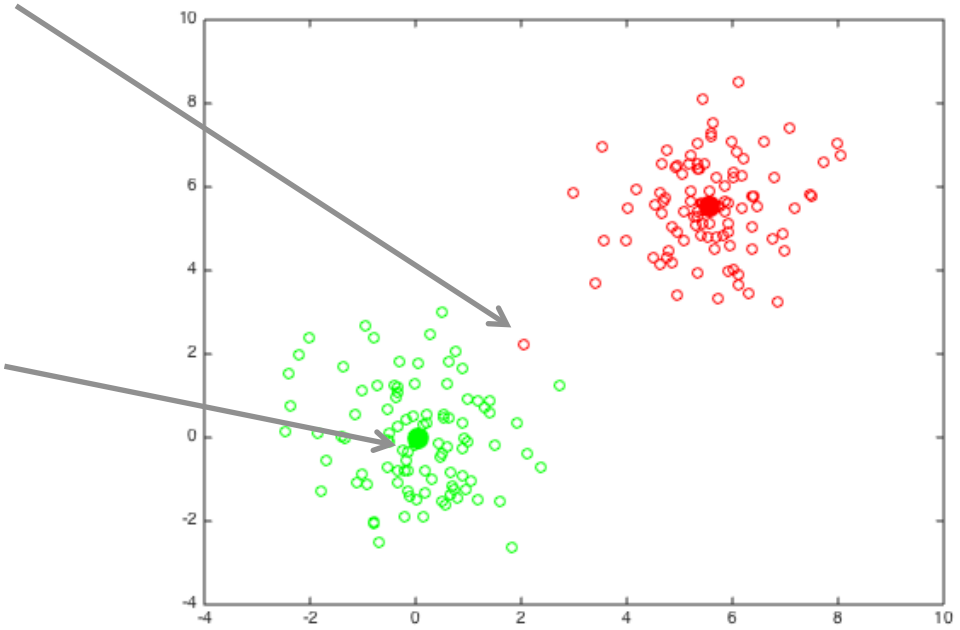
Let's go through the k-means algorithm first

Algorithm: iterative procedure

1) Pick k random points as initial cluster centers
2) Measure distance between all points and the cluster centers – assign points to nearest cluster
3) Computer cluster means

Doesn't look right

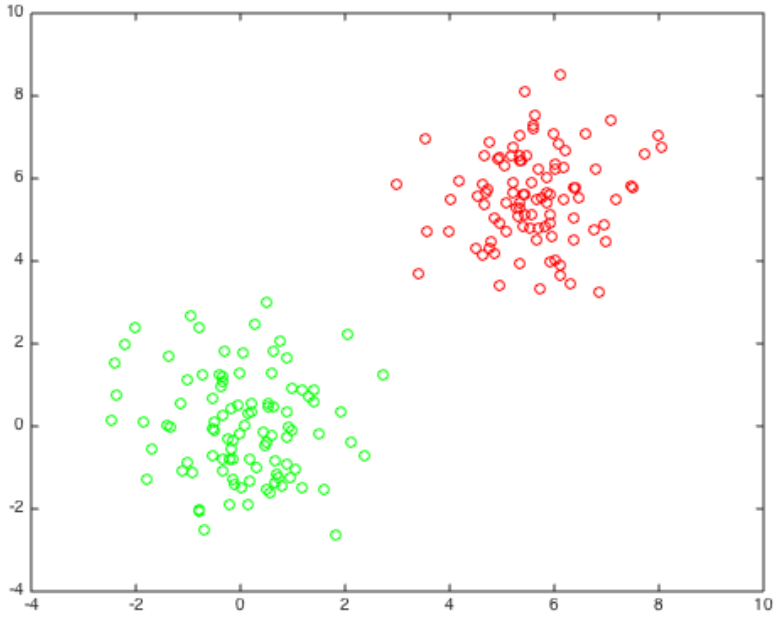
Cluster mean



Let's go through the k-means algorithm first

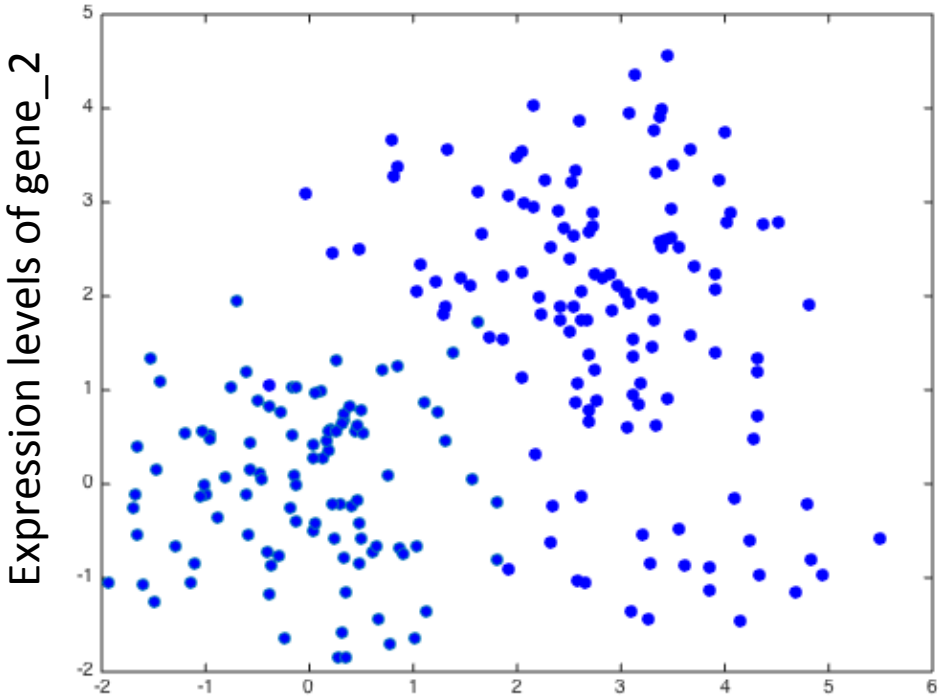
Algorithm: iterative procedure

1) Pick k random points as initial cluster centers
2) Measure distance between all points and the cluster centers – assign points to nearest cluster
3) Computer cluster means
4) Reassign points to cluster based on distance
   - If not change from previous assignment, stop, else to go step 3



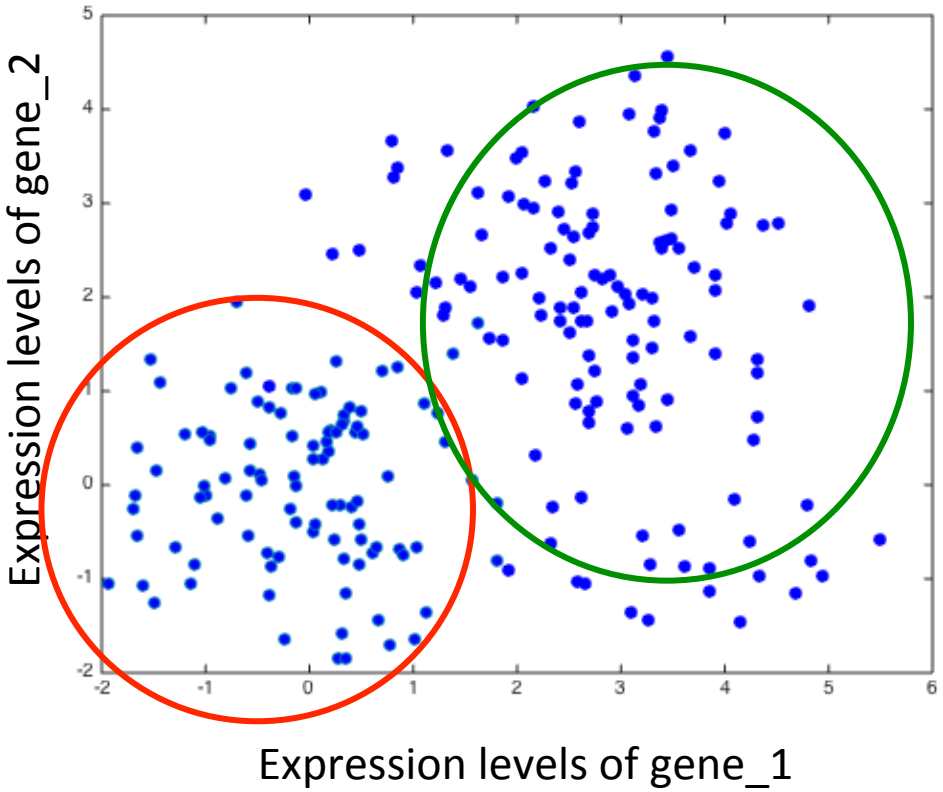
Listen 1: toy examples are “easy”

How many clusters are there?



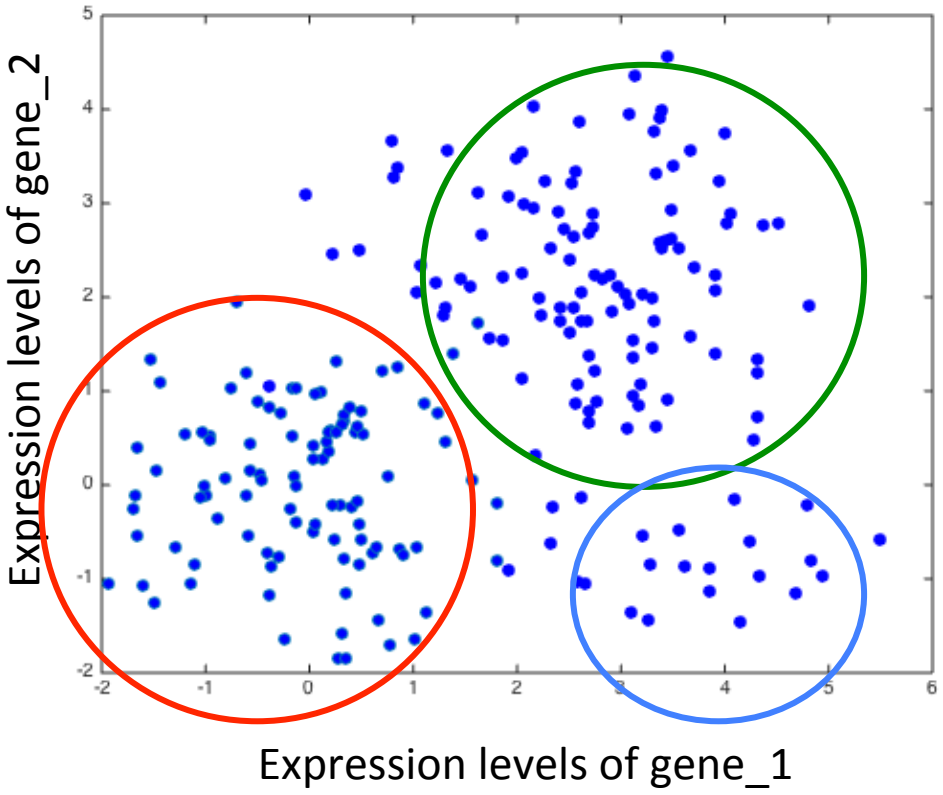
Listen 1: toy examples are “easy”

How many clusters are there?



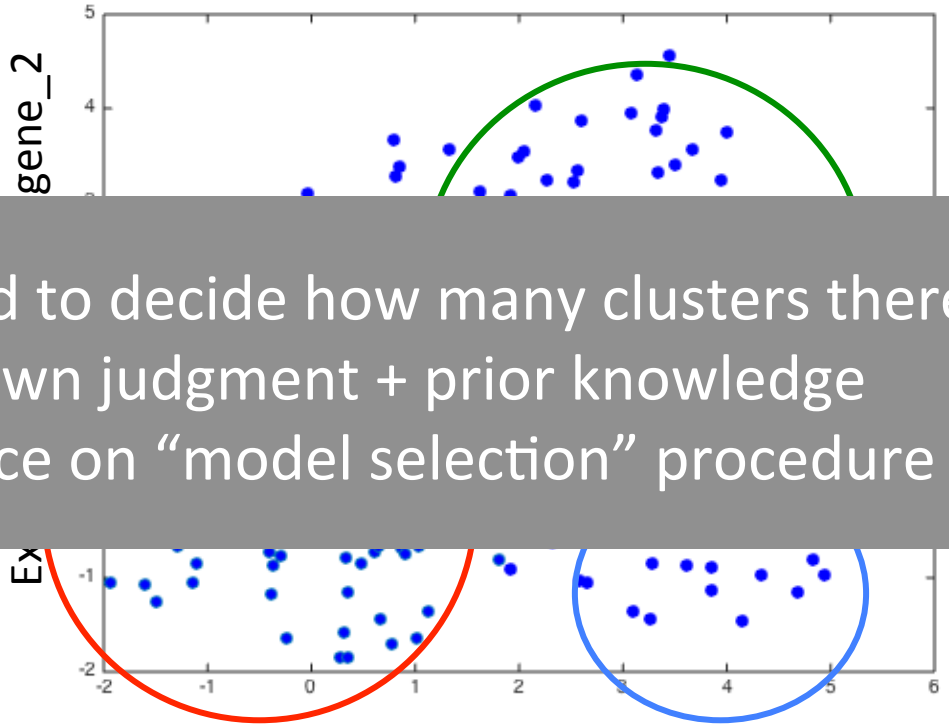
Listen 1: toy examples are “easy”

How many clusters are there?



Listen 1: toy examples are “easy”

How many clusters are there?



Expression levels of gene_1

We need to decide how many clusters there are first:

* Your own judgment + prior knowledge
* Reliance on “model selection” procedure

K-means objective function (formula/equation)

Objective function: minimize the average squared **Euclidean distance** of objects from their assigned cluster centers. A **cluster center** (or centroid) is defined as the mean of objects in the given cluster.

* N objects, each have p attributes: $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \ldots, \vec{x}_n'\}$

* k-means objective function:

$$J = \sum_{i=1}^{n} \sum_{k=1, i \in k}^{K} \| \vec{x}_1 - \vec{\mu}_k \|$$

Cluster center for cluster k

Euclidian distance between x_i and u_k

Algorithms: k-means

Note that

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d(\mathbf{X}_i, \mathbf{X}_j) = \sum_{r=1}^{K}\sum_{i\in\mathcal{C}_r}\sum_{j=1}^{n} d(\mathbf{X}_i, \mathbf{X}_j)$$

$$= \sum_{r=1}^{K}\sum_{i\in\mathcal{C}_r}\left[\sum_{j\in\mathcal{C}_r} d(\mathbf{X}_i, \mathbf{X}_j) + \sum_{j\notin\mathcal{C}_r} d(\mathbf{X}_i, \mathbf{X}_j)\right]$$

$$= \sum_{r=1}^{K}\sum_{i,j\in\mathcal{C}_r} d(\mathbf{X}_i, \mathbf{X}_j) + \sum_{r=1}^{K}\sum_{i\in\mathcal{C}_r}\sum_{j\notin\mathcal{C}_r} d(\mathbf{X}_i, \mathbf{X}_j)$$

$$T = W + B$$

When $d(\mathbf{X}_i, \mathbf{X}_j) = \|\mathbf{X}_i - \mathbf{X}_j\|^2$

$$W = \sum_{r=1}^{K} \sum_{i,j \in C_r} \|\mathbf{X}_i - \mathbf{X}_j\|^2 = \sum_{r=1}^{K} \sum_{i \in C_r} \|\mathbf{X}_i - \bar{\mathbf{X}}_r\|^2$$

- Given $\overline{\mathbf{X}}_1, \overline{\mathbf{X}}_2, \ldots, \overline{\mathbf{X}}_K$, the minimum of *W* is attained assigning $\mathbf{X}_i$ to the cluster C_r with the closest mean ($\overline{\mathbf{X}}_r$).
- Given $C_1, C_2, \ldots, C_K$, the minimum of *W* is attained estimating the center of the cluster with its sample mean $\overline{\mathbf{X}}_r$.

$$\min_{\hat{\mu}_1, \ldots, \hat{\mu}_K} \sum_{r=1}^{k} \sum_{i \in C_r} \|\mathbf{x}_i - \hat{\mu}_r\|^2 \longrightarrow \hat{\mu}_r = \bar{\mathbf{X}}_r = \frac{1}{n_r} \sum_{i \in C_r} \mathbf{x}_i$$

The K-means algorithm

Algorithm: iterative procedure

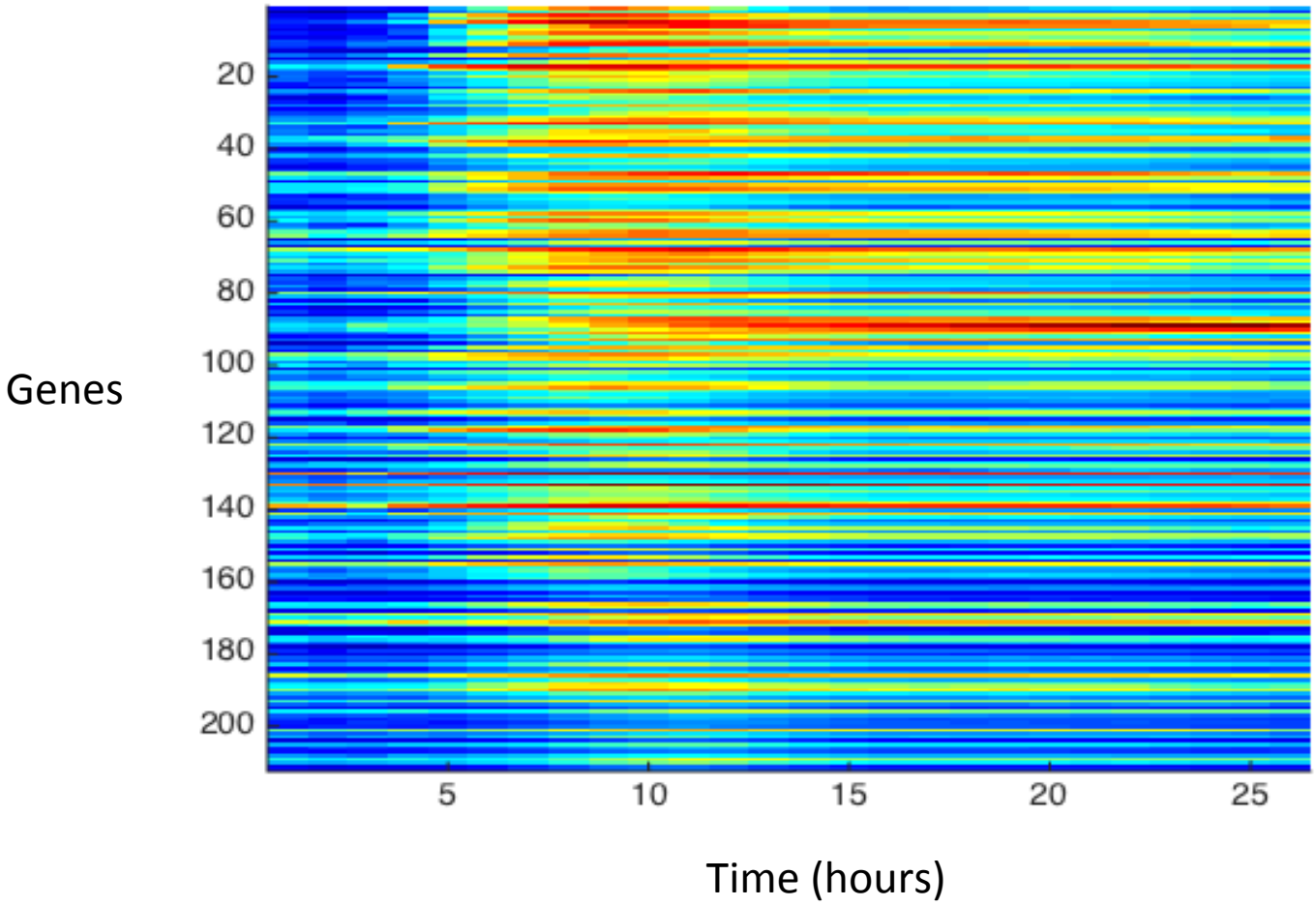
Initialize: randomly assign **k** cluster centers
Iterate:

1) Measure distance between all points and the cluster centers – assign points to nearest cluster
2) Update cluster centers
3) Re-assign points to clusters
4) Stop if there are no (or minimal) re-assignments otherwise go to step 1.

Timing patterns for IFN induced genes in CD19^{+} Bcells

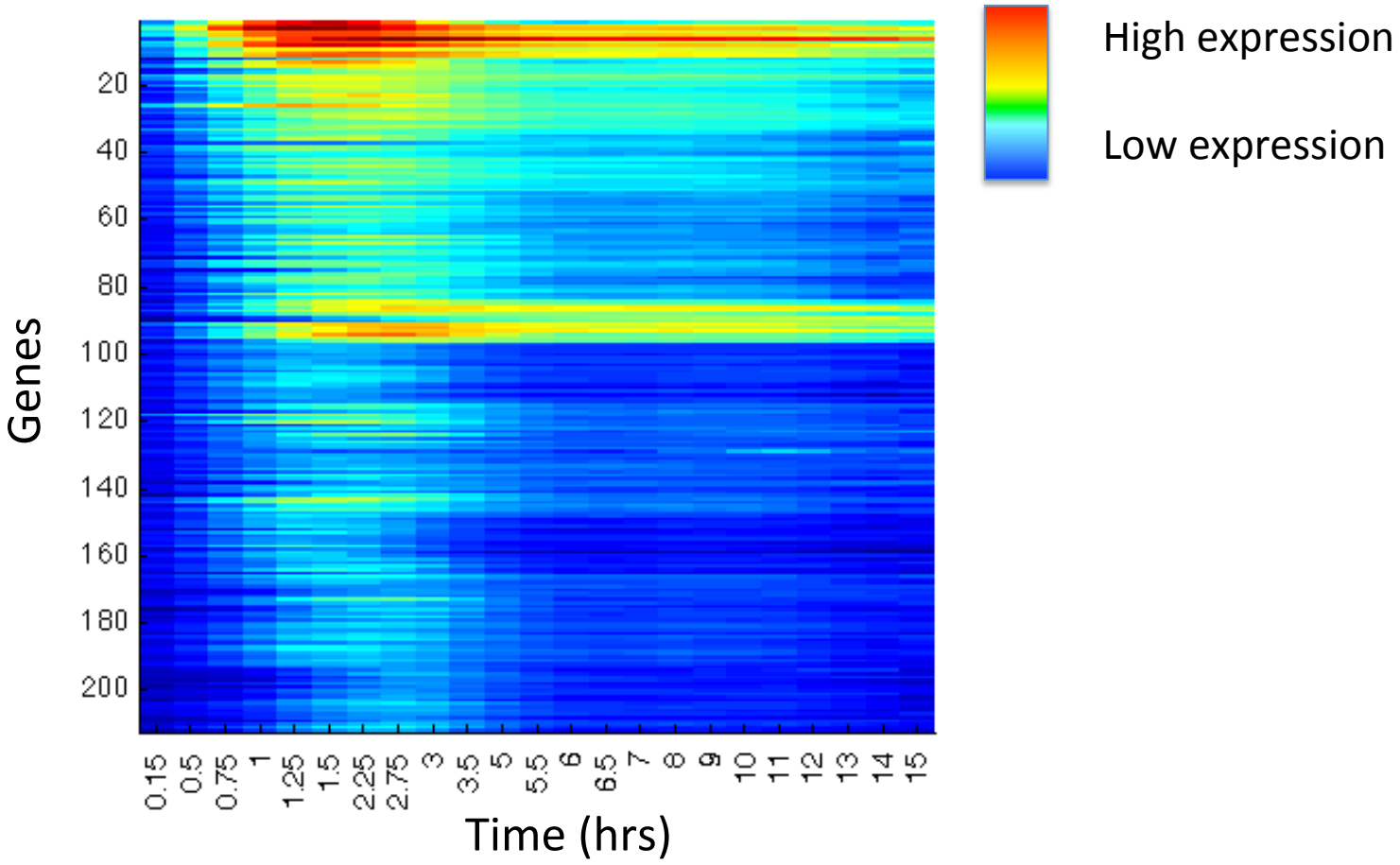
High expression

Low expression



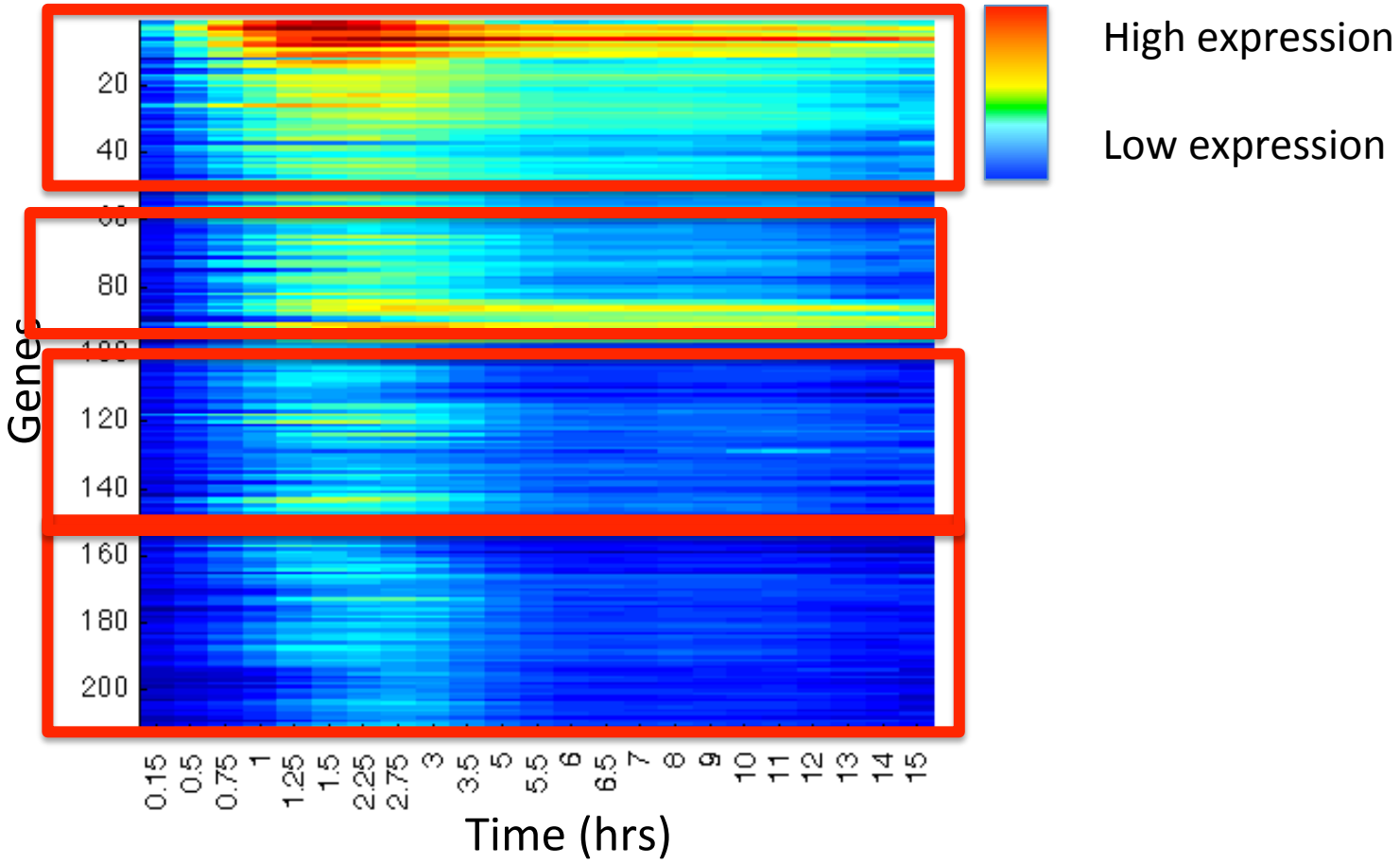
Timing patterns for IFN induced genes in CD19^{+} Bcells

Application of k-means clustering with k=4



Timing patterns for IFN induced genes in CD19^{+} Bcells

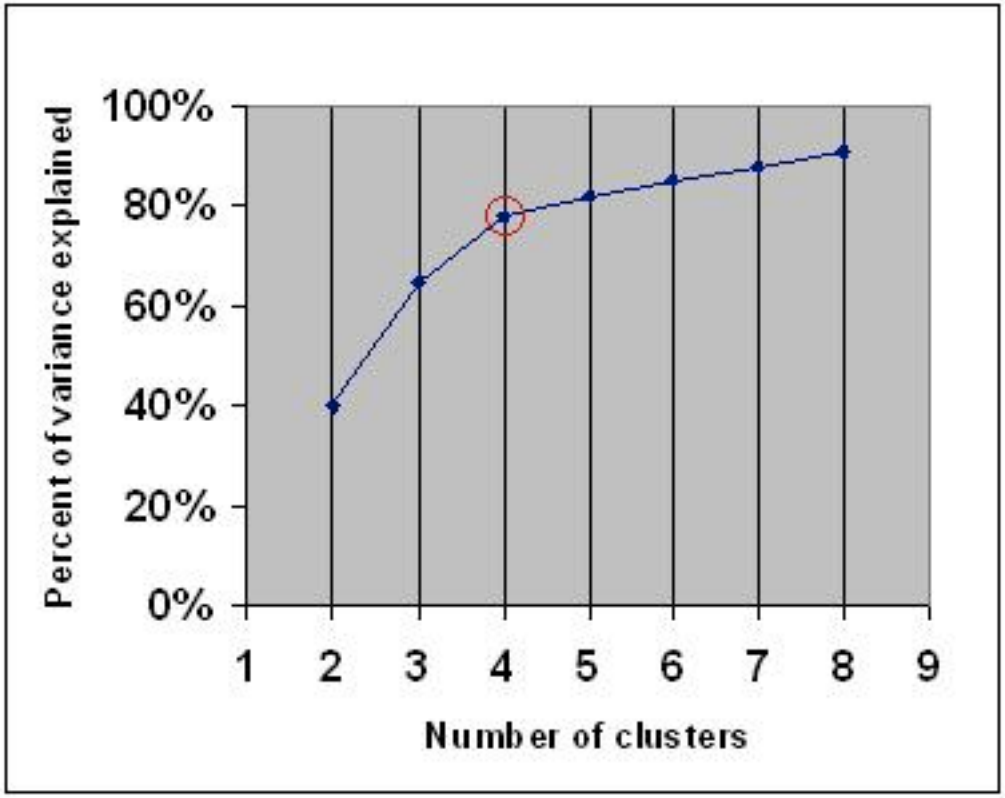
Application of k-means clustering with k=4



How do you determine *k* (number of clusters)?

Note: maximizing the clustering likelihood/objective will not be informative→ each object should be in its own cluster. Therefore, need an algorithm that takes into account the "cost" of additional clusters.

- Prior knowledge
- The "elbow method"



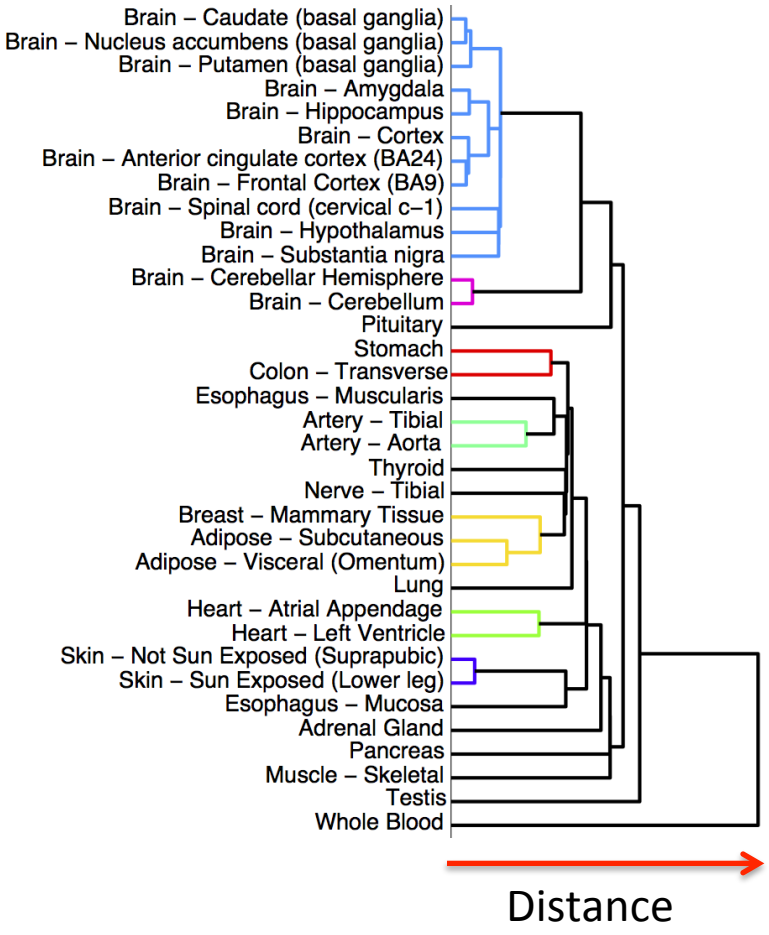
How do you determine *k* (number of clusters)?

Note: maximizing the clustering likelihood/objective will not be informative→ each object should be in its own cluster. Therefore, need an algorithm that takes into account the "cost" of additional clusters.

- Prior knowledge
- The "elbow method"
- Information Criteria Approach: AIC or BIC
- Silhouette method
- The Gap Statistics
- Cross-validation

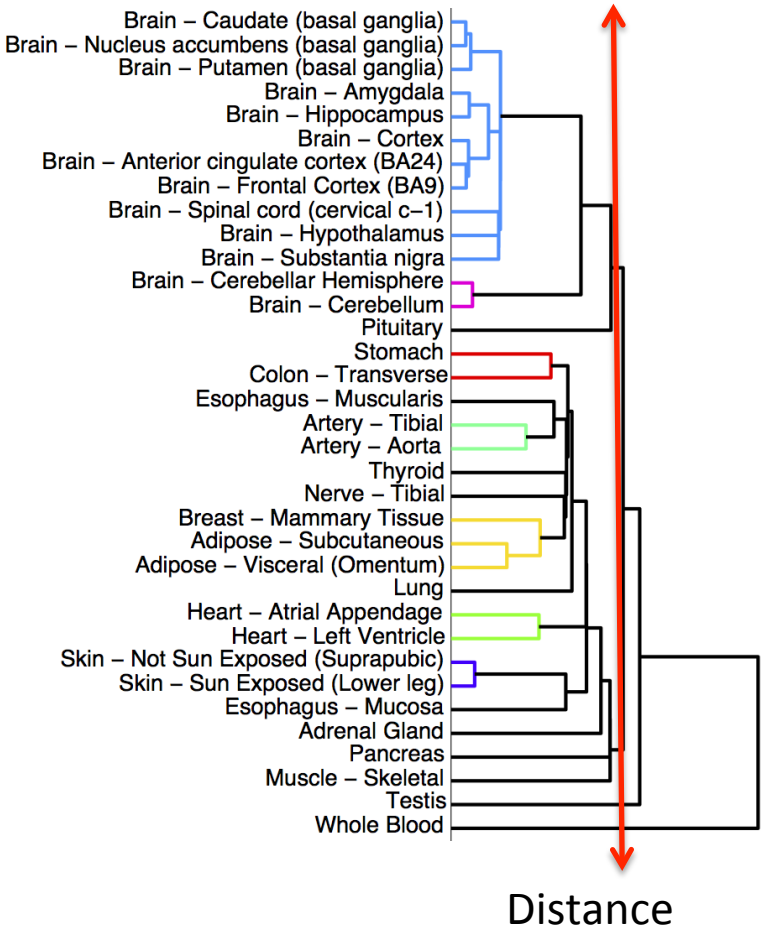
Hierarchical Agglomerative clustering

A clustering approach for revealing hierarchical relationships between objects



Hierarchical Agglomerative clustering

A clustering approach for revealing hierarchical relationships between objects



What is Clustering?

- "Clustering" Colloquially means placing/grouping a set of objects into groups/clusters.

- Clustering is a formal **problem** in Computer Science and in Statistics, with formal definitions and "solutions".

- Clustering gives you a way to summarize your high-dimensional data → so a summary statistics that is 'high dimensional' itself

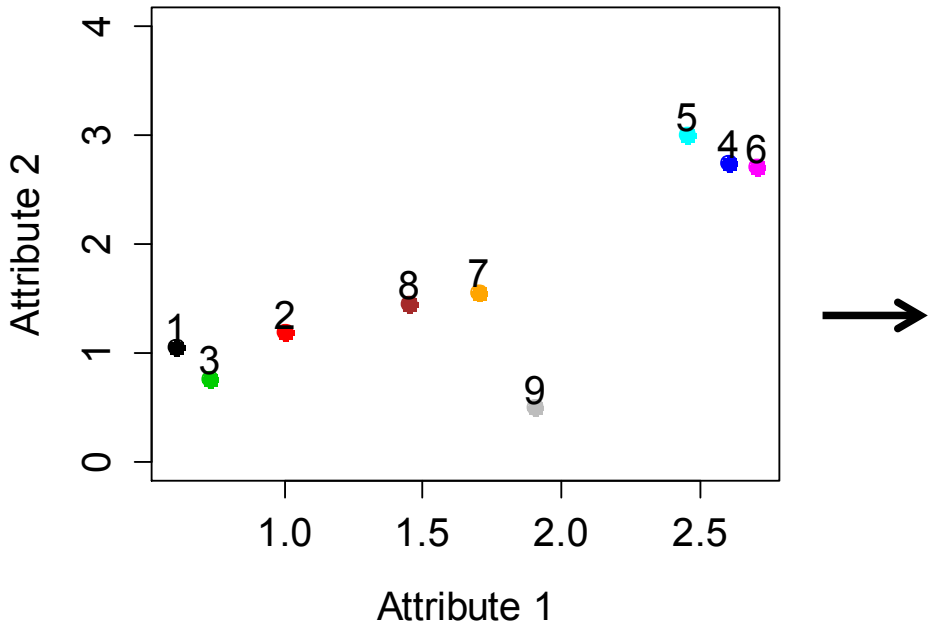
- Therefore, think of it as a *random* processes – if you want to interpret it you must investigate the distribution of your statistics

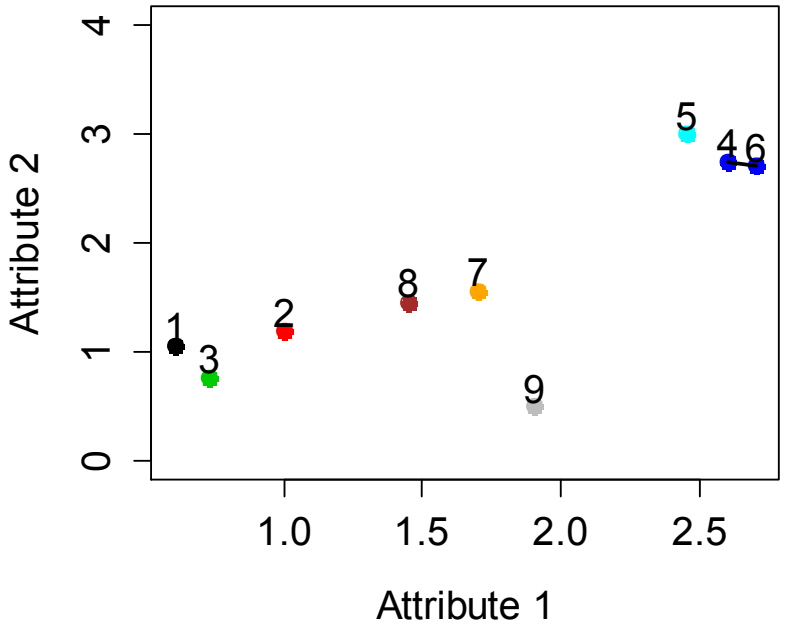
c_i

Algorithms: Hierarchical

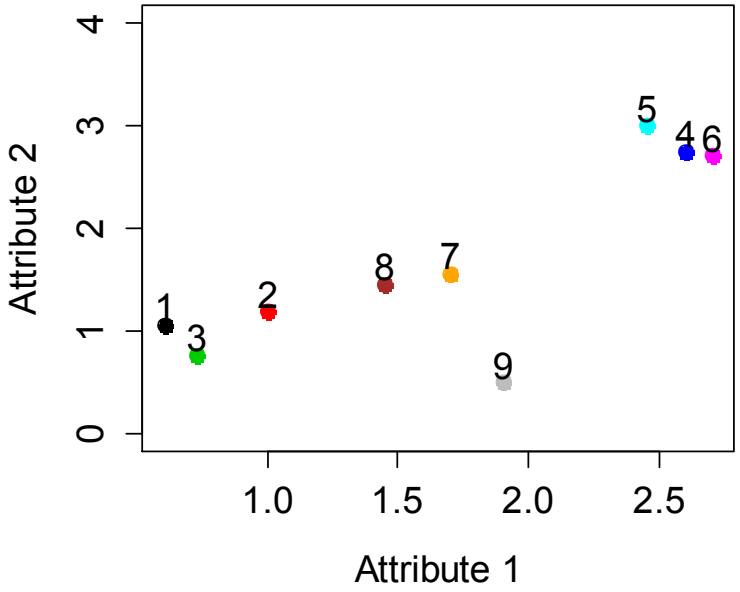
Given *N objects* with *H attributes* and a *distance metric*:

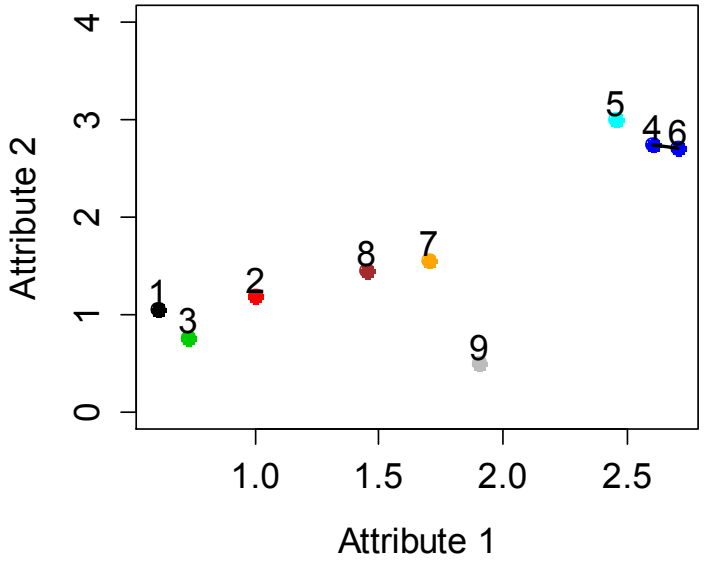
1. Assign each object to a cluster and compute the pairwise distances between all clusters
2. Find the "closest" pair of *clusters* and *merge them* into a single cluster
3. Compute new distances between clusters
4. Repeat steps 2 and 3 until all objects belong to a single cluster.






```
> round(dist(a, method='euclidean'),2)
     1    2    3    4    5    6    7    8
2 0.41
3 0.32 0.50
4 2.61 2.23 2.72
5 2.67 2.32 2.81 0.29
6 2.66 2.28 2.76 0.11 0.39
7 1.20 0.79 1.25 1.49 1.62 1.52
8 0.93 0.52 0.99 1.73 1.84 1.77 0.27
9 1.41 1.13 1.20 2.35 2.55 2.34 1.07 1.05
```

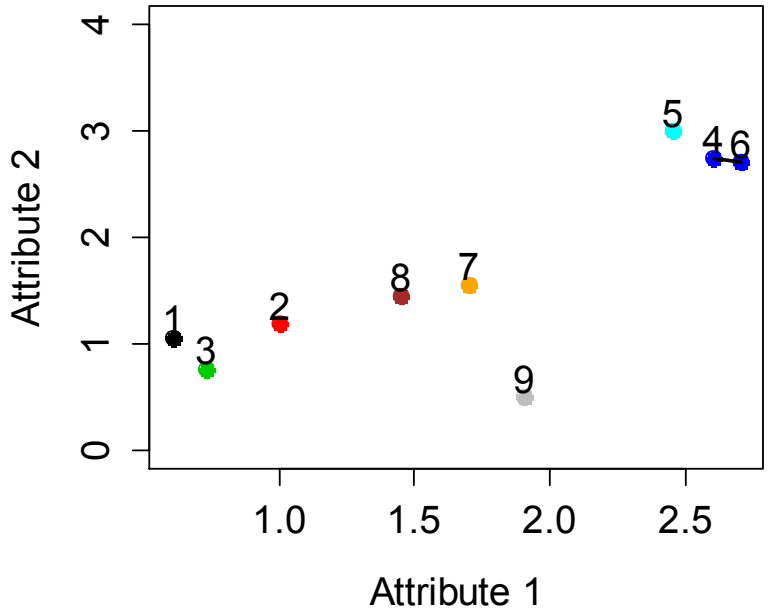


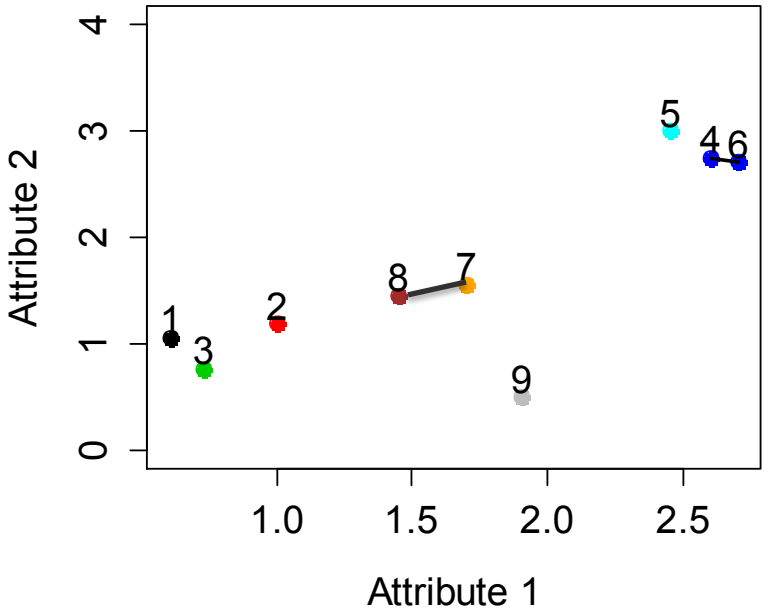



```
> round(dist(a, method='euclidean'),2)
     1    2    3    4    5    6    7    8
2 0.41
3 0.32 0.50
4 2.61 2.23 2.72
5 2.67 2.32 2.81 0.29
6 2.66 2.28 2.76 0.11 0.39
7 1.20 0.79 1.25 1.49 1.62 1.52
8 0.93 0.52 0.99 1.73 1.84 1.77 0.27
9 1.41 1.13 1.20 2.35 2.55 2.34 1.07 1.05
```

→ You can define the cluster "centroids" using:

- Single linkage
- Average linkage
- Complete linkage

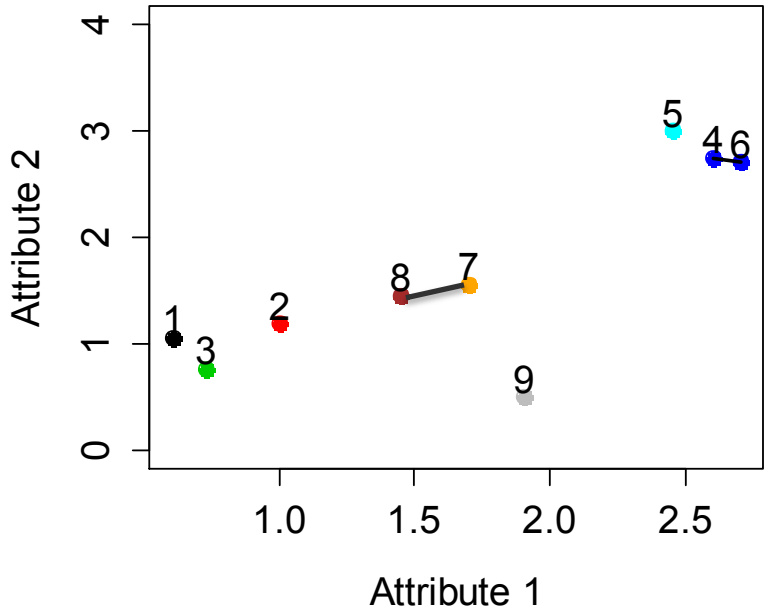


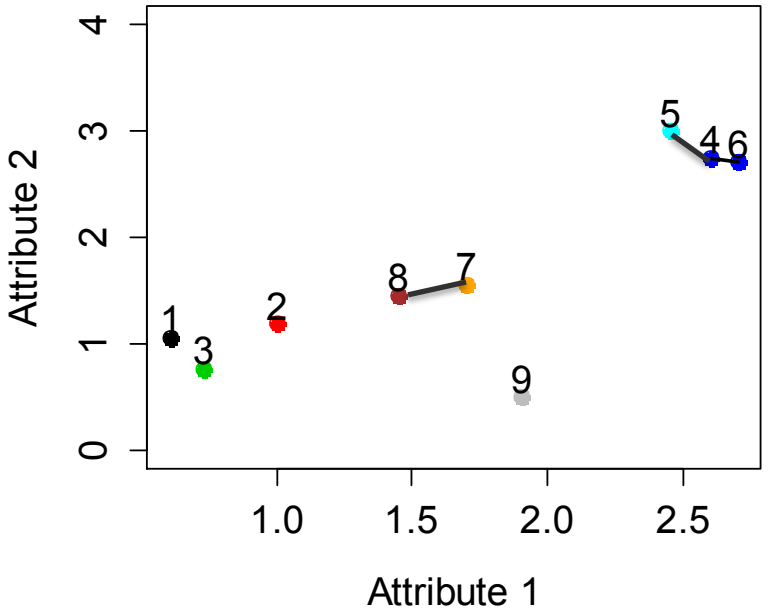



```
> round(dist(a, method='euclidean'),2)
     1    2    3    4    5    6    7    8
2 0.41
3 0.32 0.50
4 2.61 2.23 2.72
5 2.67 2.32 2.81 0.29
6 2.66 2.28 2.76 0.11 0.39
7 1.20 0.79 1.25 1.49 1.62 1.52
8 0.93 0.52 0.99 1.73 1.84 1.77 0.27
9 1.41 1.13 1.20 2.35 2.55 2.34 1.07 1.05
```

→ You can define the cluster "centroids" using:

- Single linkage
- Average linkage
- Complete linkage

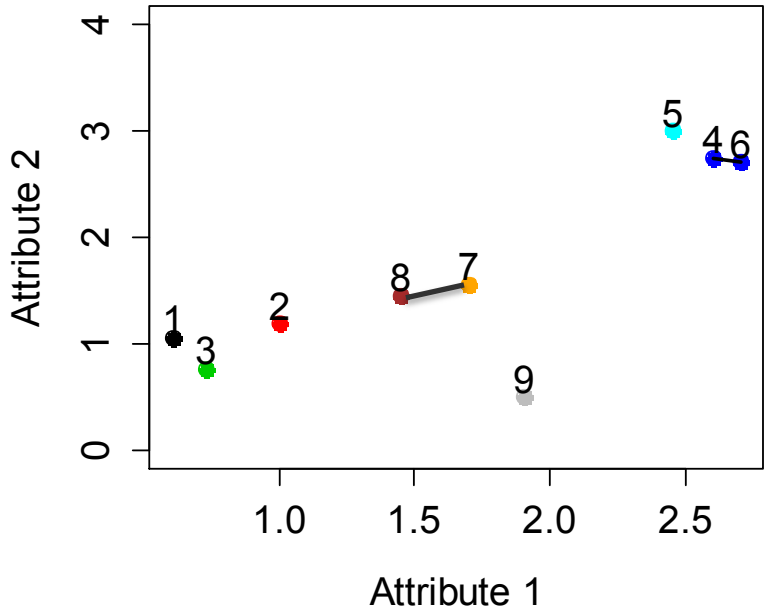


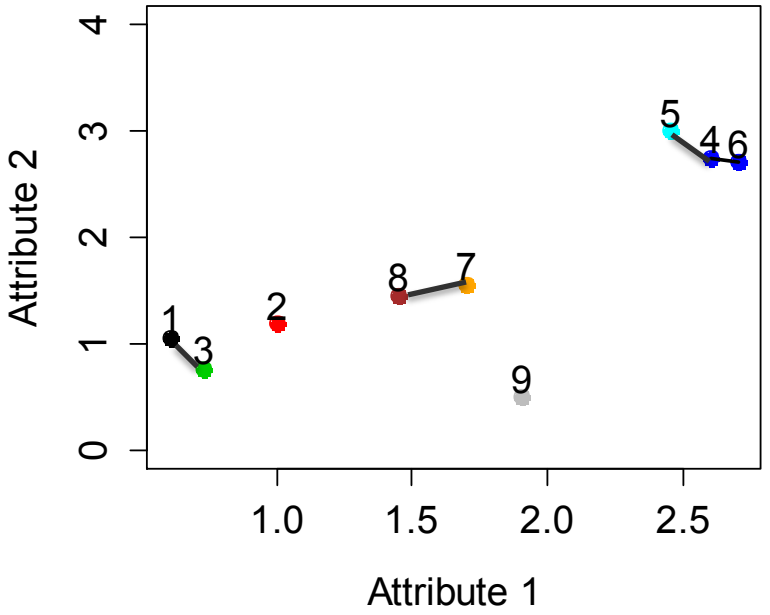



```
> round(dist(a, method='euclidean'),2)
     1    2    3    4    5    6    7    8
2 0.41
3 0.32 0.50
4 2.61 2.23 2.72
5 2.67 2.32 2.81 0.29
6 2.66 2.28 2.76 0.11 0.39
7 1.20 0.79 1.25 1.49 1.62 1.52
8 0.93 0.52 0.99 1.73 1.84 1.77 0.27
9 1.41 1.13 1.20 2.35 2.55 2.34 1.07 1.05
```

→ You can define the cluster "centroids" using:

- Single linkage
- Average linkage
- Complete linkage





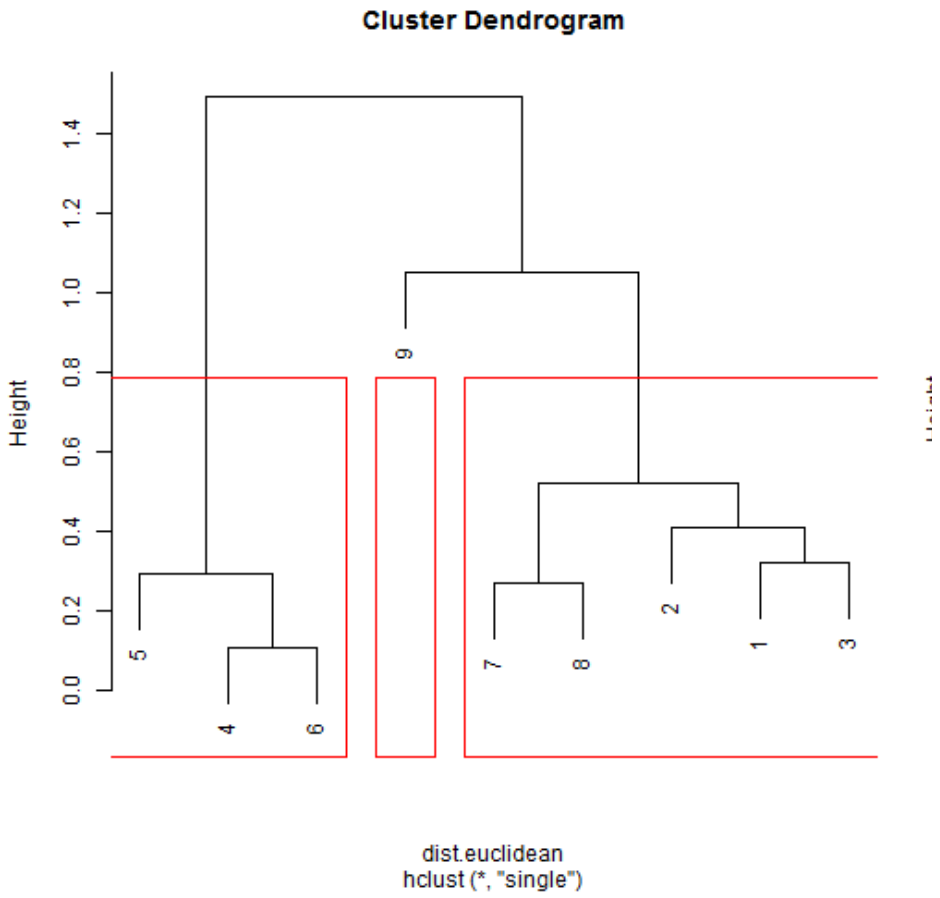

```
> round(dist(a, method='euclidean'),2)
     1    2    3    4    5    6    7    8
2 0.41
3 0.32 0.50
4 2.61 2.23 2.72
5 2.67 2.32 2.81 0.29
6 2.66 2.28 2.76 0.11 0.39
7 1.20 0.79 1.25 1.49 1.62 1.52
8 0.93 0.52 0.99 1.73 1.84 1.77 0.27
9 1.41 1.13 1.20 2.35 2.55 2.34 1.07 1.05
```

→ You can define the cluster "centroids" using:

- Single linkage
- Average linkage
- Complete linkage

Single Linkage

```r
# Dendogram
dist.euclidean = dist(a, method = "euclidean")

# Single
ex1.hcS <- hclust(dist.euclidean, method = "single")
plot(ex1.hcS)

# identify 3 clusters
ex1.hcS.3 <- rect.hclust(ex1.hcS, k = 3)
```

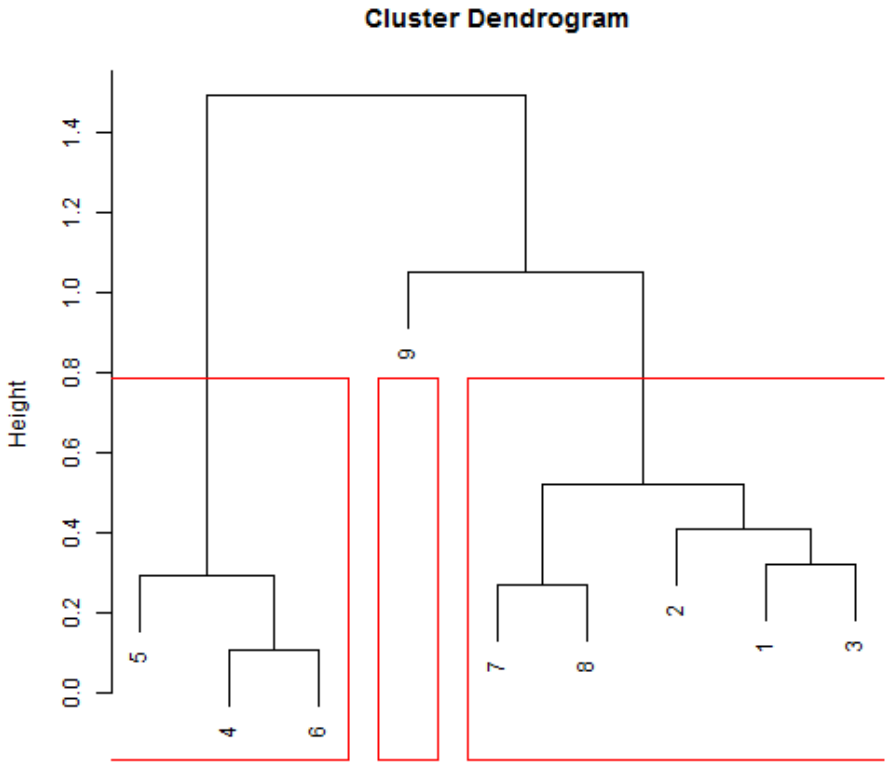


Agglomerative clustering

- **Single linkage**: The distance between two clusters is the *minimum* distance between any two elements.
- **Complete linkage**: The distance between two clusters is the *maximum* distance between any two elements.
- **Average linkage**: The distance between two clusters is the *average* of all pairwise distances between any two objects.

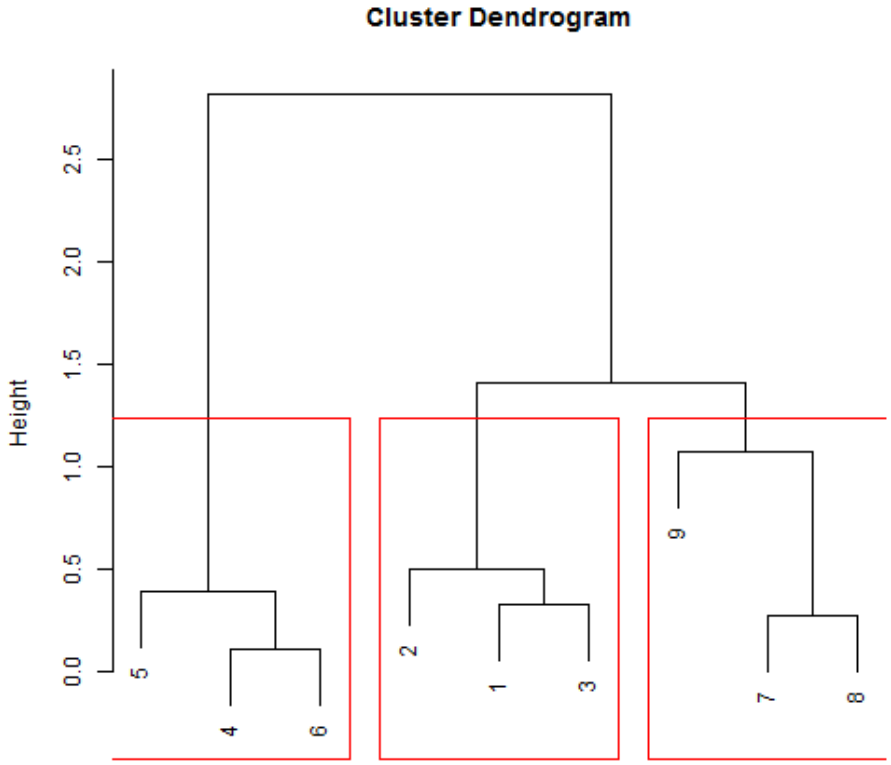
Single Linkage

```
# Dendogram
dist.euclidean = dist(a, method = "euclidean")

# Single
ex1.hcS <- hclust(dist.euclidean, method = "single")
plot(ex1.hcS)

# identify 3 clusters
ex1.hcS.3 <- rect.hclust(ex1.hcS, k = 3)
```



Complete Linkage

```
# Complete
ex1.hcC <- hclust(dist.euclidean, method = "complete")
plot(ex1.hcC)

# identify 3 clusters
ex1.hcC.3 <- rect.hclust(ex1.hcC, k = 3)
```

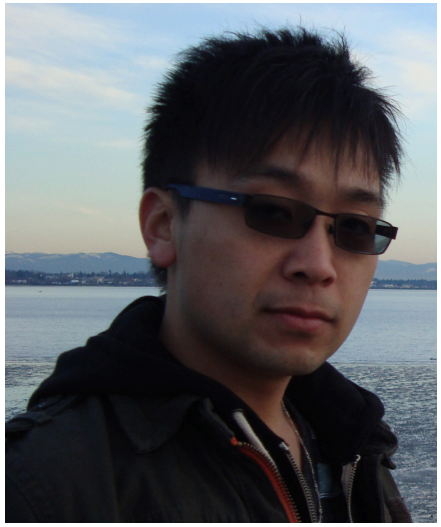


Summary & conclusions

- Many choices to make when you want to cluster a set of objects:
  - Objective, algorithm, **attributes/features**, distance metric, number of clusters.
- Not possible to say which method is the best. It all depends on data and goal.
- Clustering is very powerful, but reckless application leads to misguided conclusions.
- Don’t do surgery before you are comfortable with using Band-Aids (L. Wasserman)

Talk announcement

Title: Multiomic Association Analysis with Kernel Machines



Date: Tuesday, March 8, 2016

Location: Room 4192, Earth Sciences Building (2207 Main Mall)

Time: 11:00am - 12:00pm