Clustering in Bioinformatics

Overview

- ☐ Define the clustering problem
- ☐ Motivation: gene expression and microarrays
- ☐ Types of clustering
- ☐ Clustering algorithms
- ☐ Other applications of clustering

The clustering problem

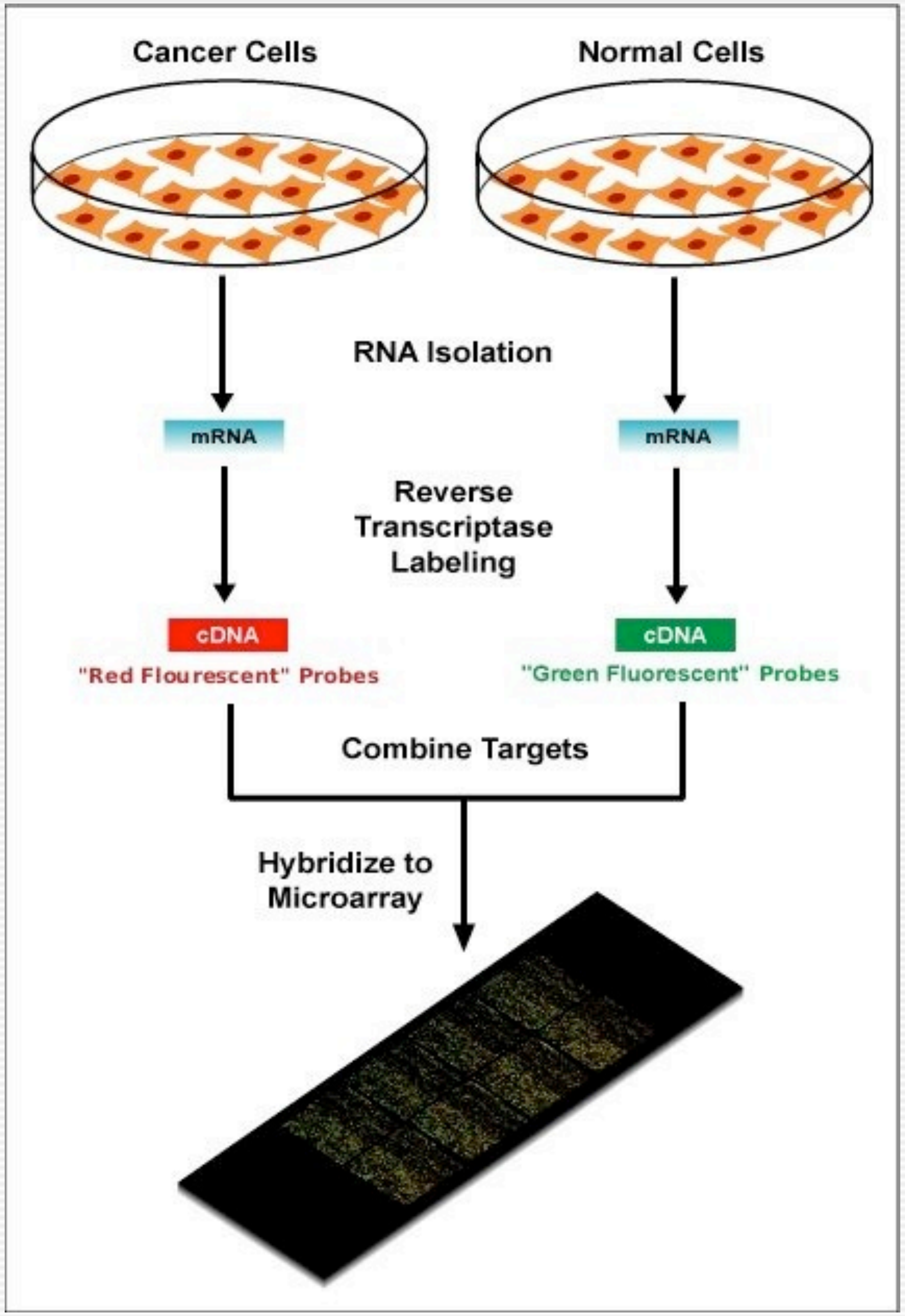
- Motivation: Find patterns in a sea of data
- Input:
  - A (large) number of datapoints: **N**
  - A measure of distance between any two data points $\mathbf{d}_{ij}$
- Output:
  - Groupings (**clustering**) of the elements into **K** (the number can be user-specified or automatically determined) 'similarity' classes
  - Sometimes there is also an objective measure that the obtained clustering seeks to minimize.

A marquee application: Microarray analysis

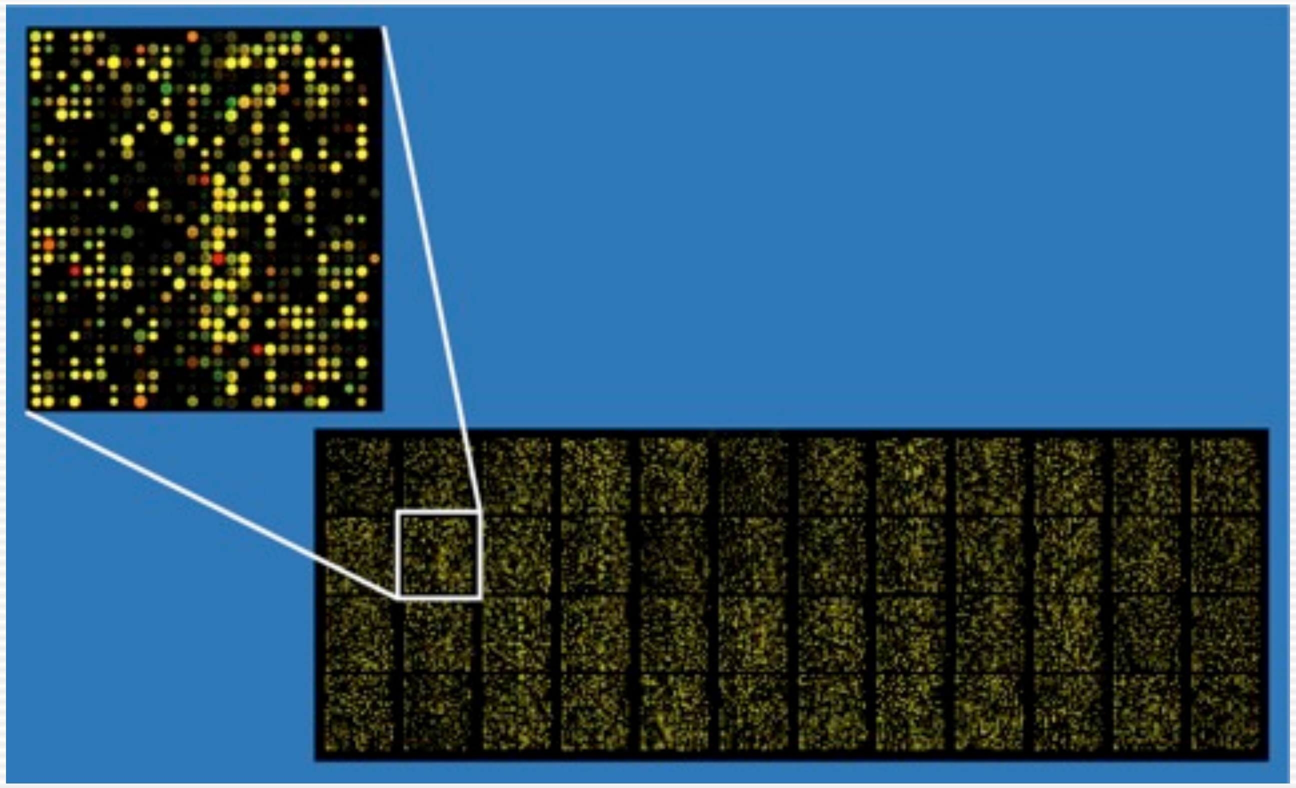
- What do newly sequenced genes do?
- Simply comparing the new gene sequences to known DNA sequences often does not necessarily reveal the function of a gene: for 40% of sequenced genes, functionality cannot be ascertained by only comparing to sequences of other known genes
- Genes that perform similar or complementary function to known genes (reference) will be expressed (transcribed) at high levels together with known genes
- Genes that perform antagonistic functions (e.g. down-regulation) may be expressed at high levels at an earlier or later time point when compared to known genes
- E.g. what happens to gene expression in cancer cells?
- Expression level is estimated by measuring the amount of mRNA for that particular gene
  - A gene is active if it is being transcribed
  - More mRNA usually indicates more gene activity

A microarray experiment

- Produce cDNA from mRNA (cDNA is more stable)
- Label cDNA with a fluorescent dye or biotin for detection
- Different color labels are available to compare many samples at once
- Wash cDNA over the microarray containing thousands of high density **probes** that hybridize to complementary strands in the sample and immobilize them on the surface.
- For biotin-labeled samples, stain with the biotin-specific fluorescently labeled antibody
- Read the microarray, using a laser or a high-resolution CCD
- Illumination reveals transcribed/co-expressed genes

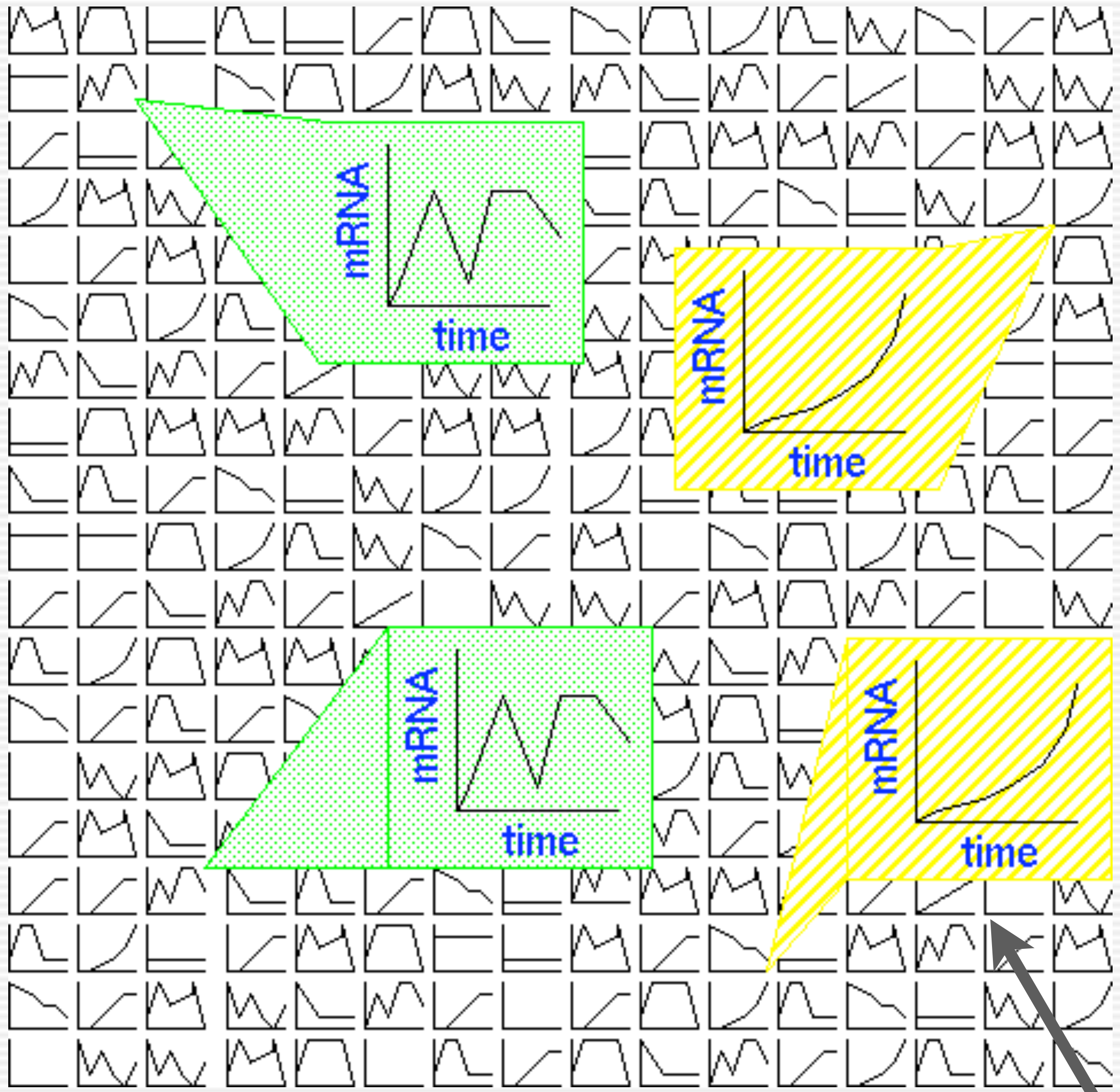


HTTP://UPLOAD.WIKIMEDIA.ORG/WIKIPEDIA/EN/C/C8/MICROARRAY-SCHEMA.JPG



HTTP://UPLOAD.WIKIMEDIA.ORG/WIKIPEDIA/COMMONS/0/0E/MICROARRAY2.GIF

- Green: expressed only in control
- Red: expressed only in an experimental cell
- Yellow: equally expressed in both samples
- Black: NOT expressed in either control or sample



- Track the sample over a period of time to observe changes in gene expression over time
- Track two samples under the same conditions to look for **differential expression**

Each box represents one gene's expression over time

Microarray Data

- Microarray data are usually transformed into a (relative, normalized) intensity matrix
- Can also be represented as a bit matrix ($\log_2$ of relative intensity)
- The intensity matrix allows biologists to infer correlations between different genes (even if they are dissimilar) and to understand how genes functions might be related
- Care must be taken to normalize the data appropriately, e.g. different time points can come from different arrays.

Gene	Time 1	Time 2	Time 3
1	10	8	10
2	10	0	9
3	4	8.5	3
4	9.5	0.5	8.5
5	4.5	8.5	3
6	10.5	9	12
7	5	8.5	11
8	2.7	8.7	2
9	9.7	2	9
10	10.2	1	9.2

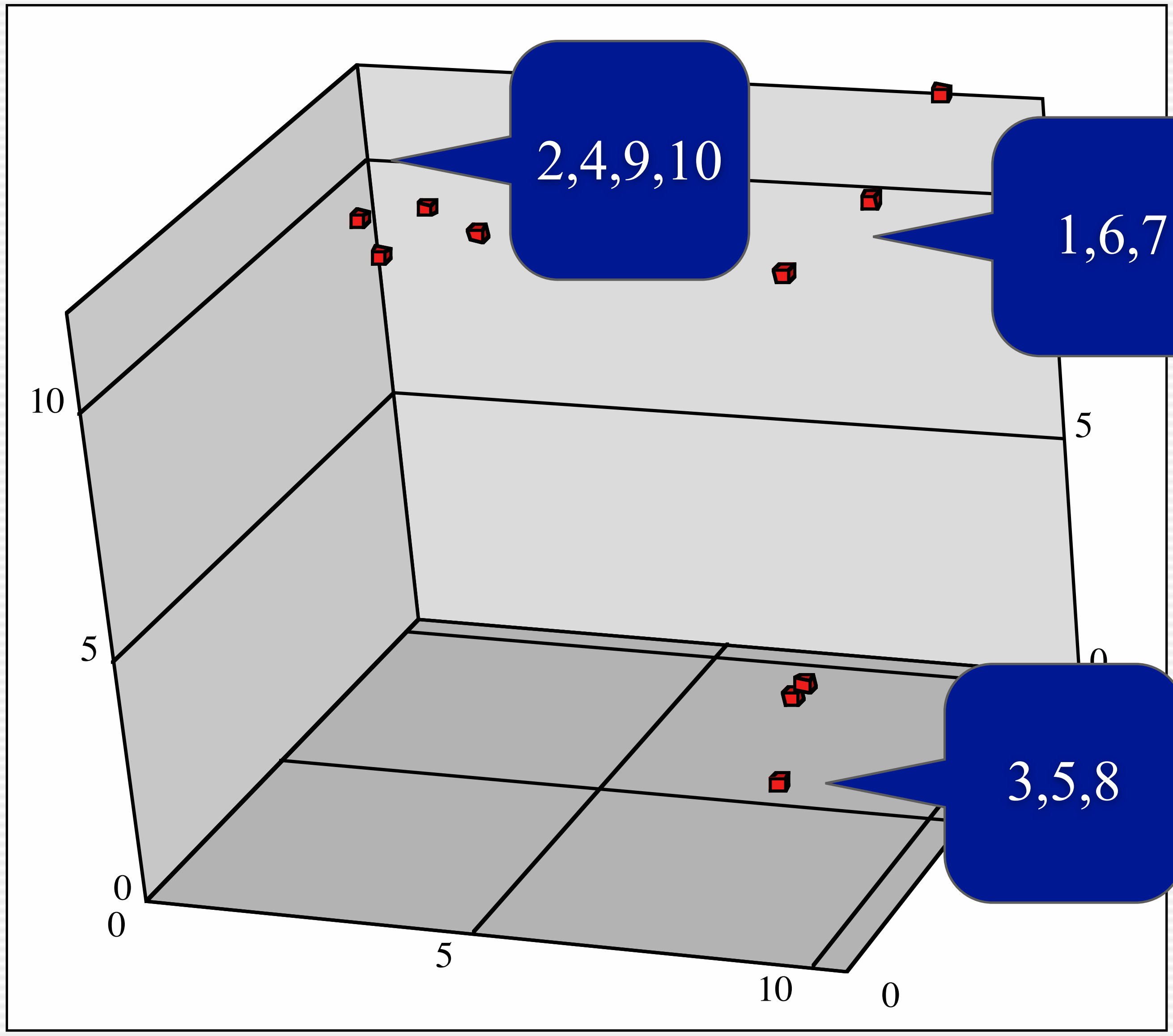
INTENSITY TABLE

- ☐ Which genes are similar?
- ☐ What defines co-expression?
- ☐ How to measure the distance/ similarity?

EUCLIDEAN DISTANCE IN D-DIMENSIONS

$$D(x,y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$$

FINDING SIMILAR GENES



	1	2	3	4	5	6	7	8	9	10
1		8.1	9.2	7.7	8.9	2.3	5.1	10.9	6.1	7.0
2	8.1		12.0	0.9	11.8	9.5	10.1	13.3	2.0	1.0
3	9.2	12.0		11.2	0.5	11.1	8.1	1.7	10.5	11.5
4	7.7	0.9	11.2		10.9	9.2	9.5	12.5	1.6	1.1
5	8.9	11.8	0.5	10.9		10.8	8.0	2.1	10.3	11.3
6	2.3	9.5	11.1	9.2	10.8		5.6	12.7	7.7	8.5
7	5.1	10.1	8.1	9.5	8.0	5.6		9.3	8.3	9.3
8	10.9	13.3	1.7	12.5	2.1	12.7	9.3		12.0	12.9
9	6.1	2.0	10.5	1.6	10.3	7.7	8.3	12.0		1.1
10	7.0	1.0	11.5	1.1	11.3	8.5	9.3	12.9	1.1	

PAIRWISE DISTANCES

	1	6	7	2	4	9	10	3	5	8
1	0.0	2.3	5.1	8.1	7.7	6.1	7.0	9.2	8.9	10.9
6	2.3	0.0	5.6	9.5	9.2	7.7	8.5	11.1	10.8	12.7
7	5.1	5.6	0.0	10.1	9.5	8.3	9.3	8.1	8.0	9.3
2	8.1	9.5	10.1	0.0	0.9	2.0	1.0	12.0	11.8	13.3
4	7.7	9.2	9.5	0.9	0.0	1.6	1.1	11.2	10.9	12.5
9	6.1	7.7	8.3	2.0	1.6	0.0	1.1	10.5	10.3	12.0
10	7.0	8.5	9.3	1.0	1.1	1.1	0.0	11.5	11.3	12.9
3	9.2	11.1	8.1	12.0	11.2	10.5	11.5	0.0	0.5	1.7
6	8.9	10.8	8.0	11.8	10.9	10.3	11.3	0.5	0.0	2.1
8	10.9	12.7	9.3	13.3	12.5	12.0	12.9	1.7	2.1	0.0

REARRANGED DISTANCES

Clustering Principles

- **Homogeneity**: elements of the same cluster are maximally close to each other
- **Separation**: elements in separate clusters are maximally far apart from each other
- One is actually implied by the other (in many cases)
- Generally speaking, this is a hard problem.

$$\min_{\text{clustering}} \left[\alpha \sum_{x,y \in \text{the same cluster}} d(x,y) - \beta \sum_{x,y \in \text{different clusters}} d(x,y) \right]$$

Relative importance

Because

$$\sum_{x,y \in \text{the same cluster}} d(x,y) + \sum_{x,y \in \text{different clusters}} d(x,y) = \sum_{x,y} d(x,y) = D = const$$

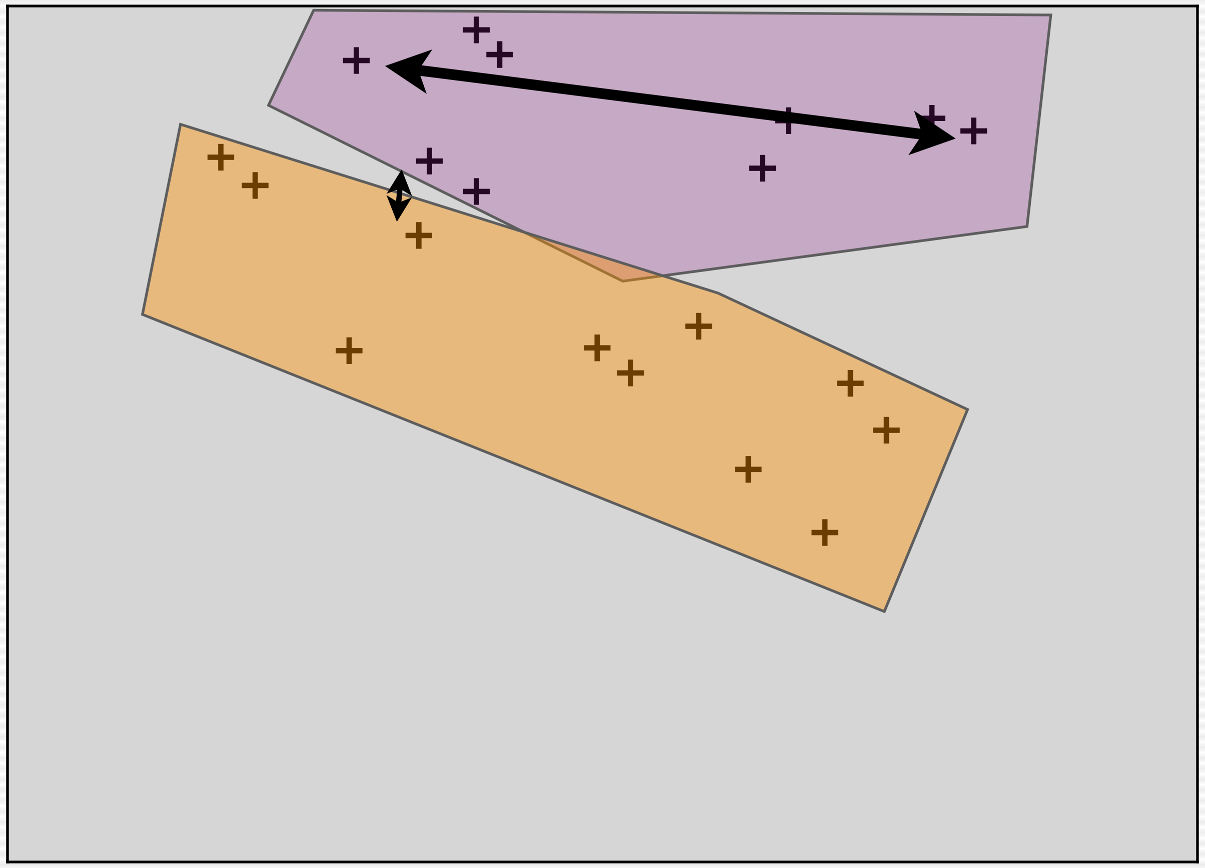
We can simplify

$$\min_{\text{clustering}} \left[\alpha \sum_{x,y \in \text{the same cluster}} d(x,y) - \beta \sum_{x,y \in \text{different clusters}} d(x,y) \right]$$

To an equivalent expression that only depends on intRA-cluster distances

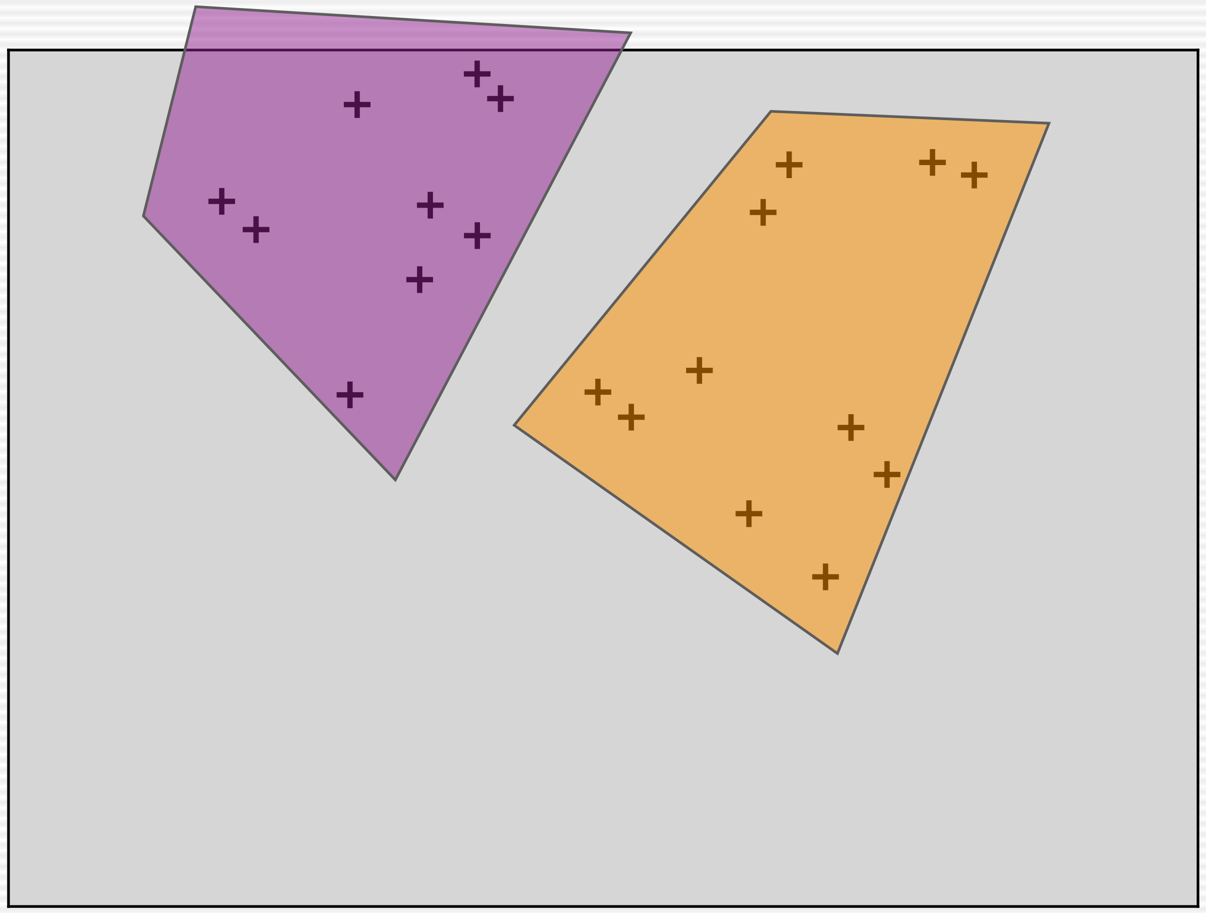
$$(\alpha + \beta) \min_{\text{clustering}} \left[\sum_{x,y \in \text{ the same cluster}} d(x,y) \right] - \beta D$$

Poor clustering example



- This clustering violates both principles:
  - Points in the same cluster are far apart
  - Points in different cluster are close together

BETTER CLUSTERING EXAMPLE



- This clustering appears sensible.
- But we need to use an **objective** metric to optimize cluster assignment.

Clustering techniques

- **Agglomerative**: Start with every element in its own cluster, and iteratively join clusters together
- **Divisive**: Start with one cluster and iteratively divide it into smaller clusters
- **Hierarchical**: Organize elements into a tree, leaves represent genes and the length of the paths between leaves represents the distances between genes. Similar genes lie within the same subtrees
- Generally, finding the exact solution to a clustering problem is NP hard.

K-MEANS CLUSTERING

- A technique to partition a set of **N** points into **K** clusters
- Each cluster is represented with a **mean** (a centroid) – hence 'K-means'
- **Input**: A set **V** with **N** points (v_1, v_2 ... v_n), the desired number of clusters **K** and a distance measure between any two points **d(v,w)**
- **Output**: A set **X** of **K** cluster centers that minimize the <u>squared error distortion</u> D(V,X) over all possible choices of X.

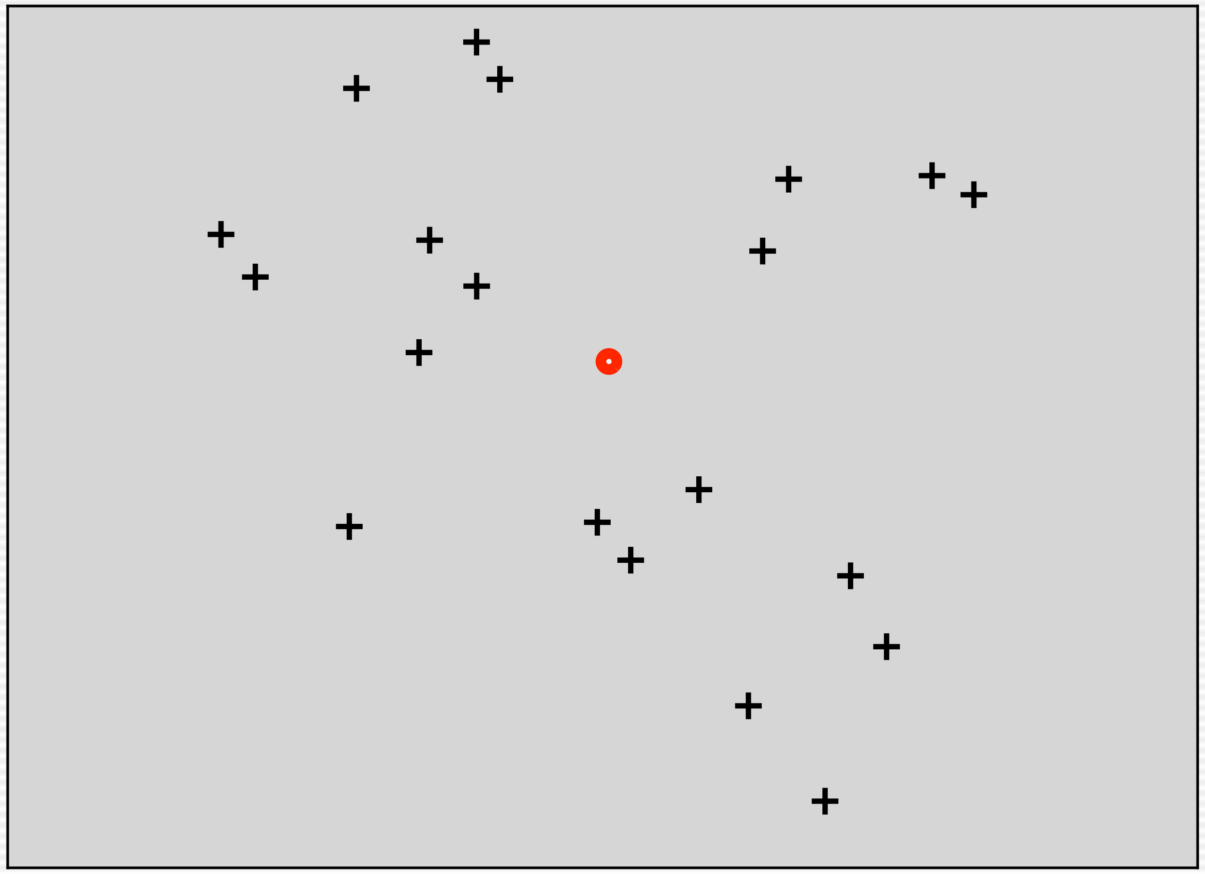
$$D(V,X) = \frac{1}{N}\sum_{i=1}^{N}\min_k d^2(v_i, x_k)$$

K-means clustering

- For K=1, the problem becomes trivial: the centroid of all points is the solution for Euclidean distances.

$$x = \frac{1}{N}\sum_i v_i$$

- For K≥2 the problem becomes NP-complete
- An efficient heuristic exists
- **Lloyd's algorithm**.



Lloyd's Algorithm

1. Arbitrarily assign the **K** cluster centers (this can significantly influence the outcome)

2. **while** cluster centers keep changing

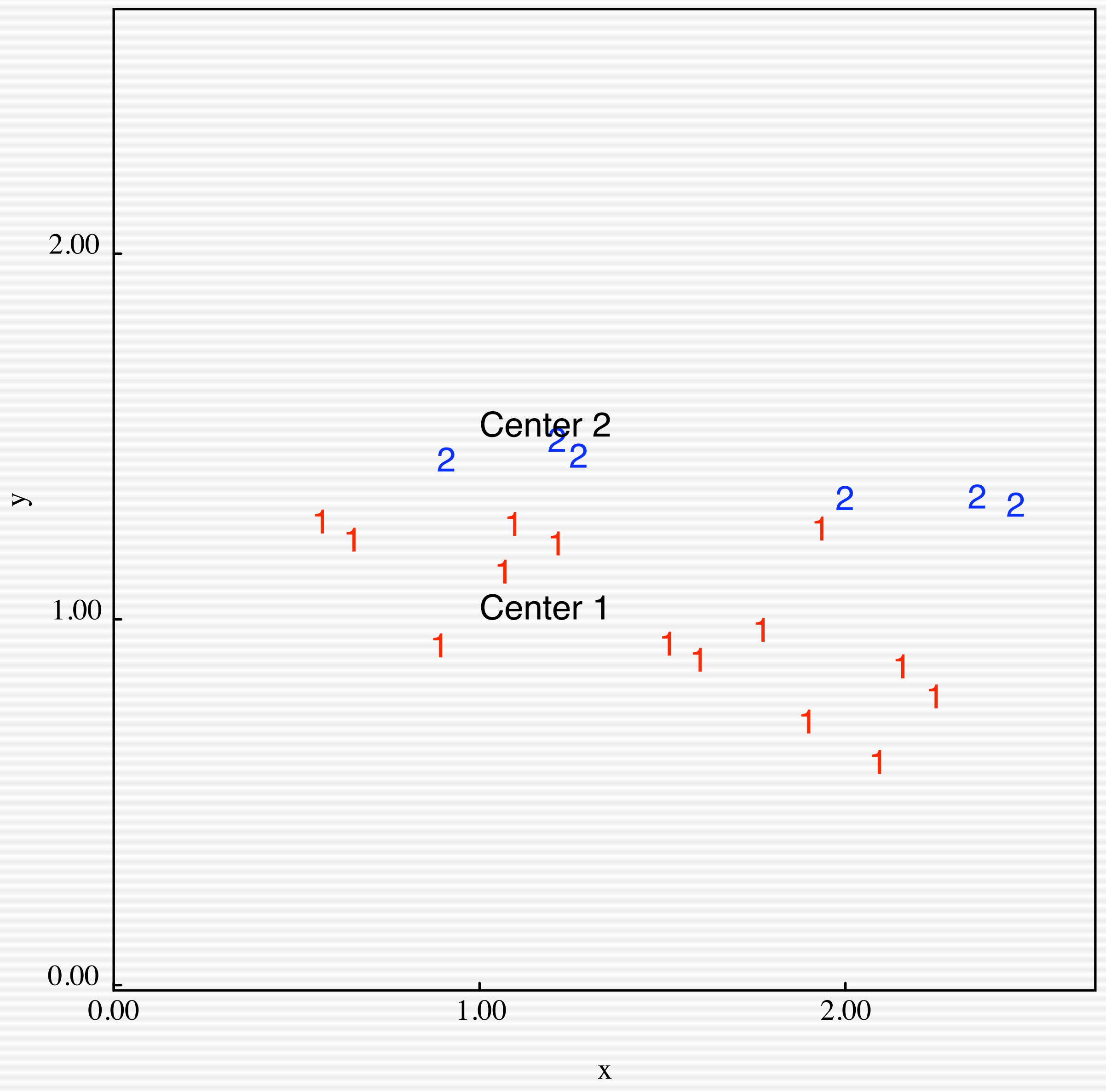
   A. Compute the distance from each data point to the current cluster center $\mathbf{C_i}$ ($1 \le i \le K$) and assign the point to the nearest cluster

   B. After the assignment of all data points, compute new centers for each cluster by taking the centroid of all the points in that cluster

3. Output cluster centers and assignments

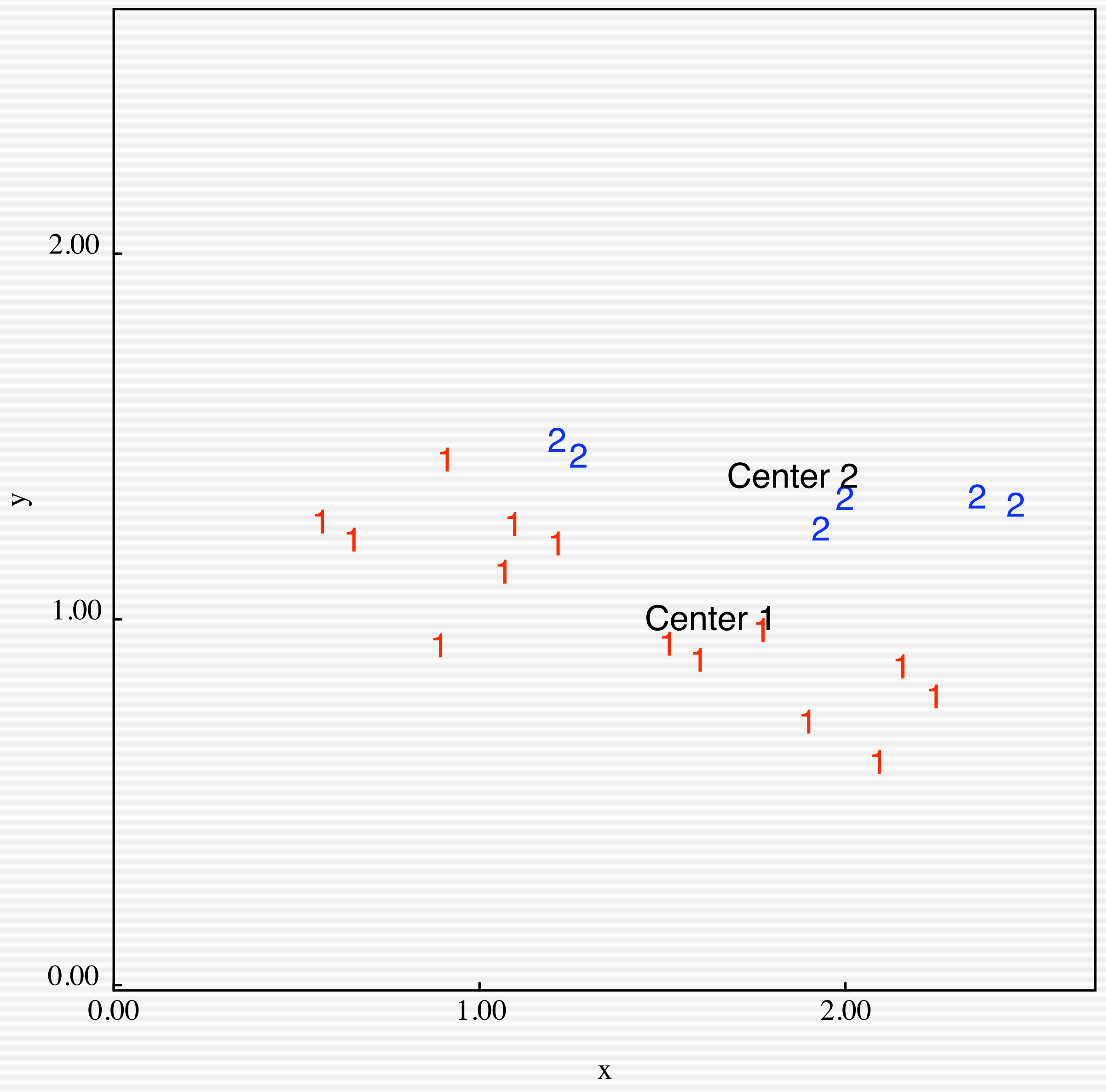
K-MEANS EXECUTION EXAMPLE

STEP 1



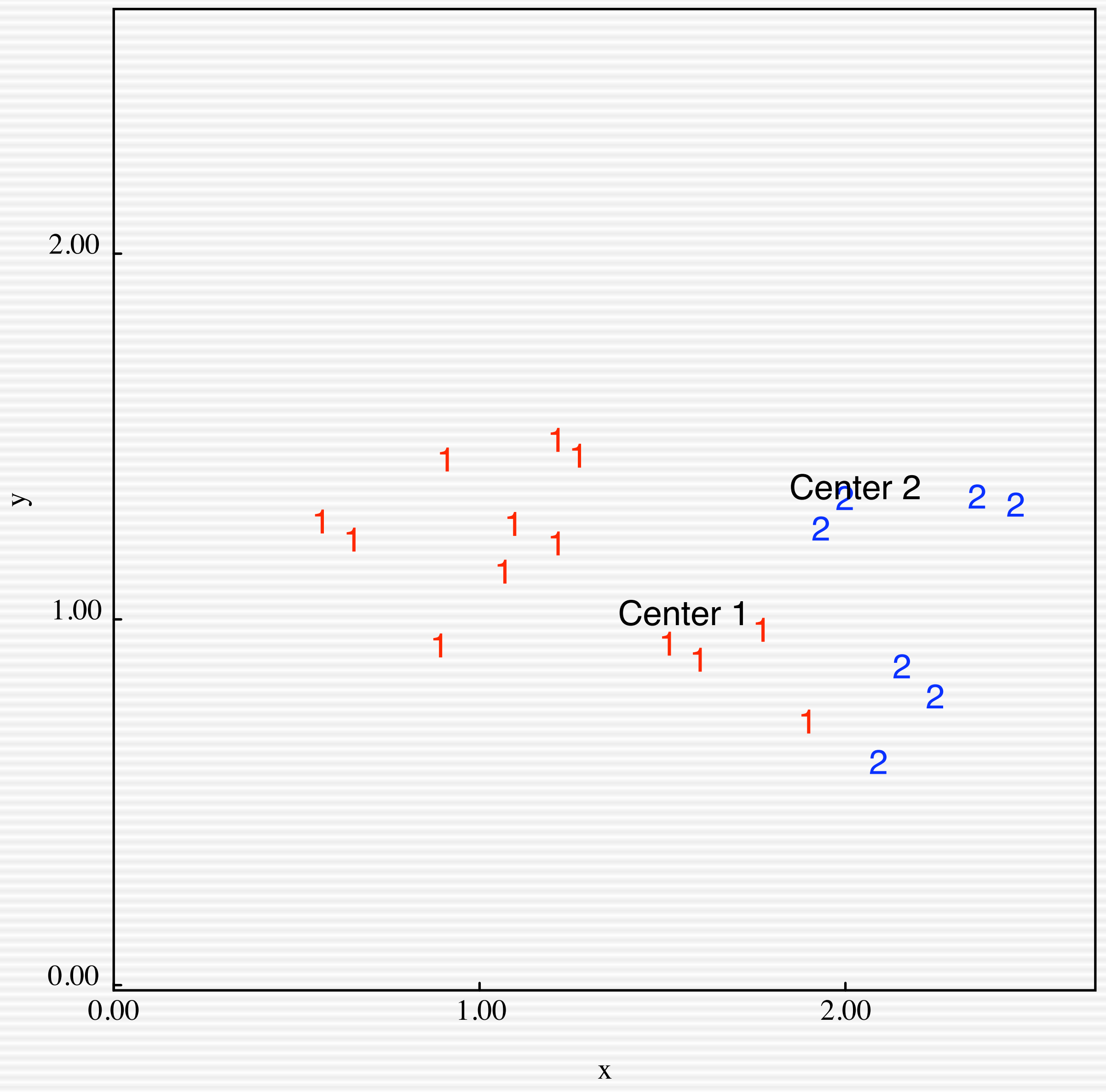
K-MEANS EXECUTION EXAMPLE

STEP 2



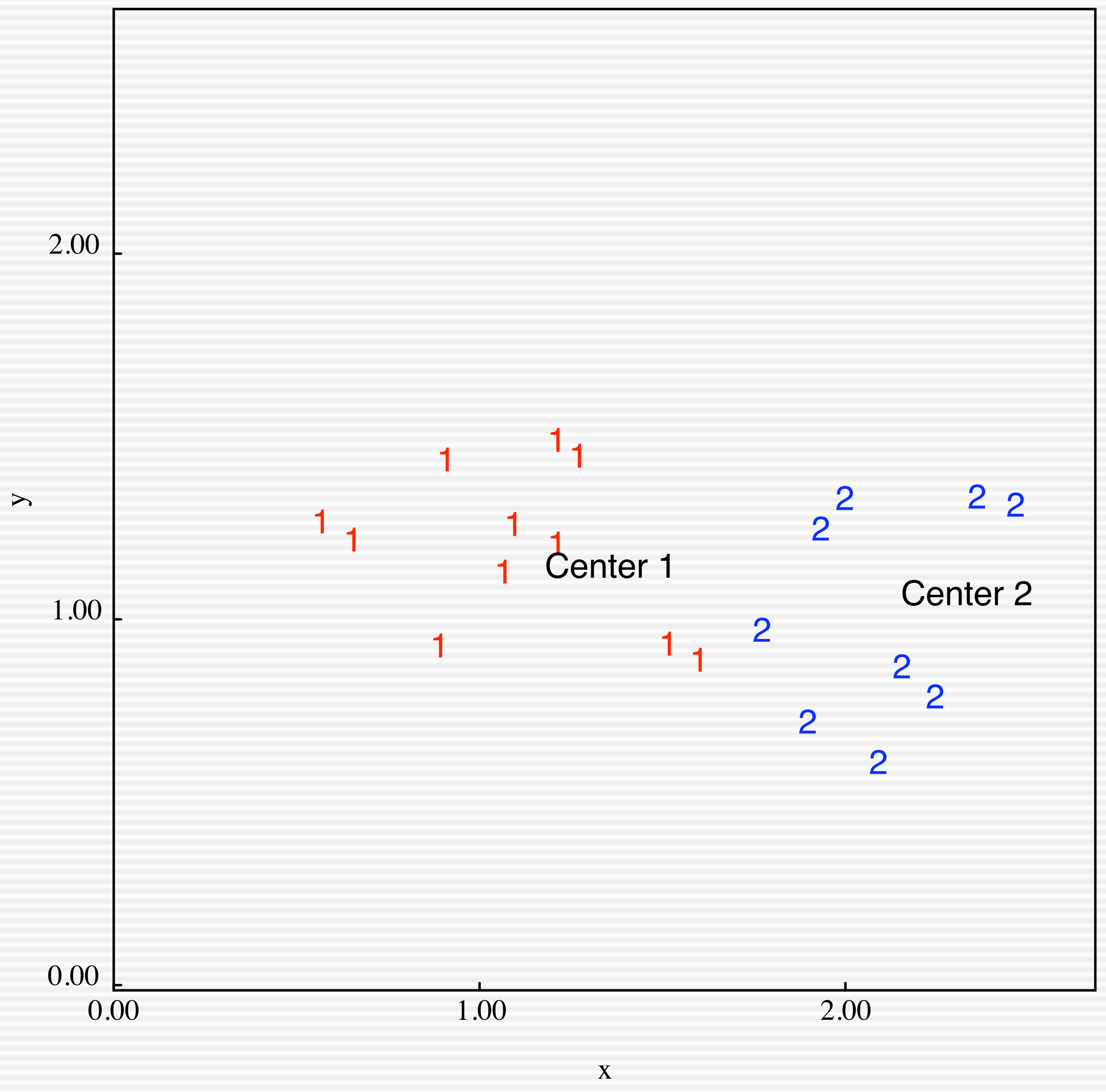
K-MEANS EXECUTION EXAMPLE

STEP 3



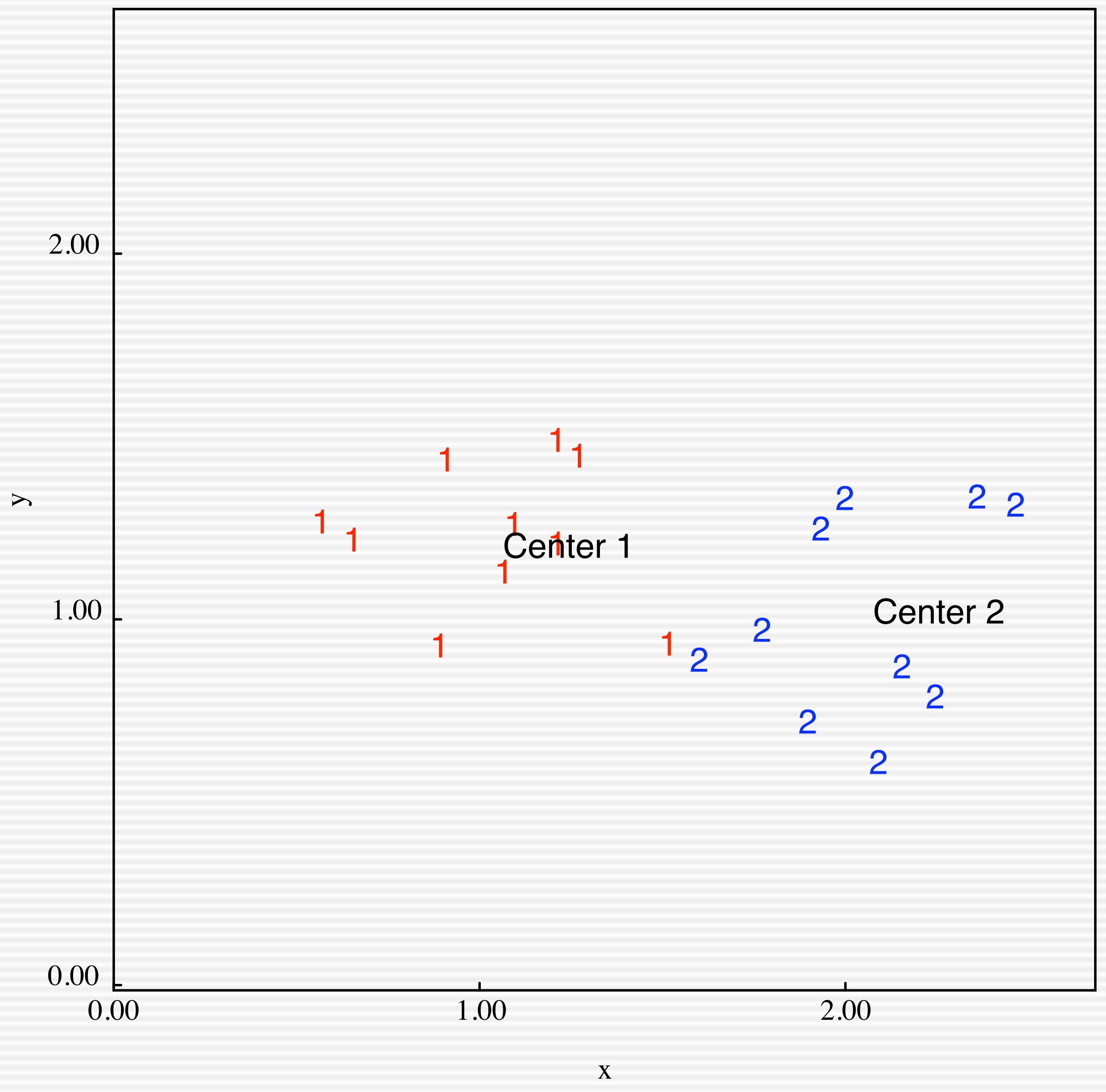
K-means execution example

Step 4



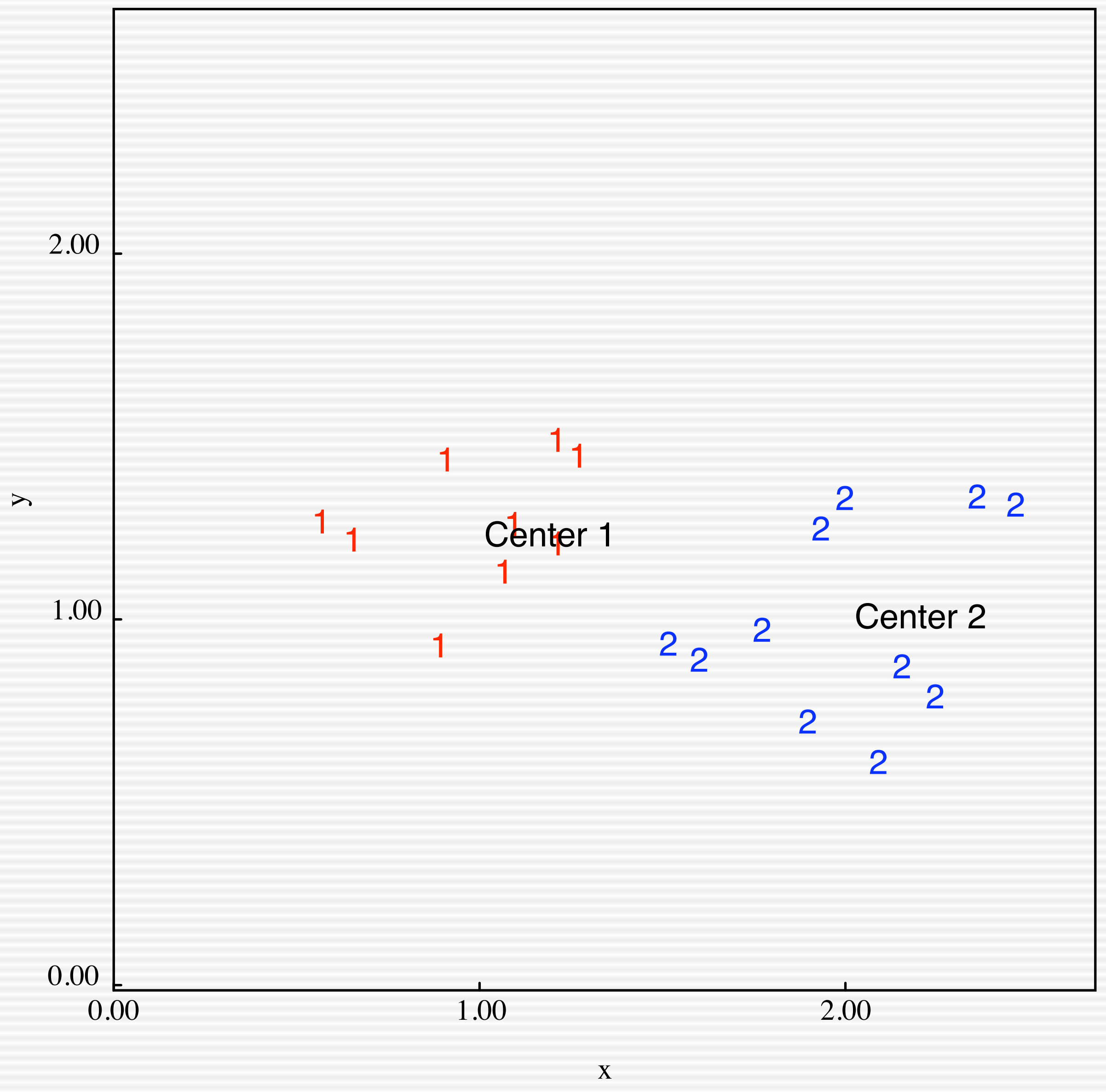
K-MEANS EXECUTION EXAMPLE

STEP 5

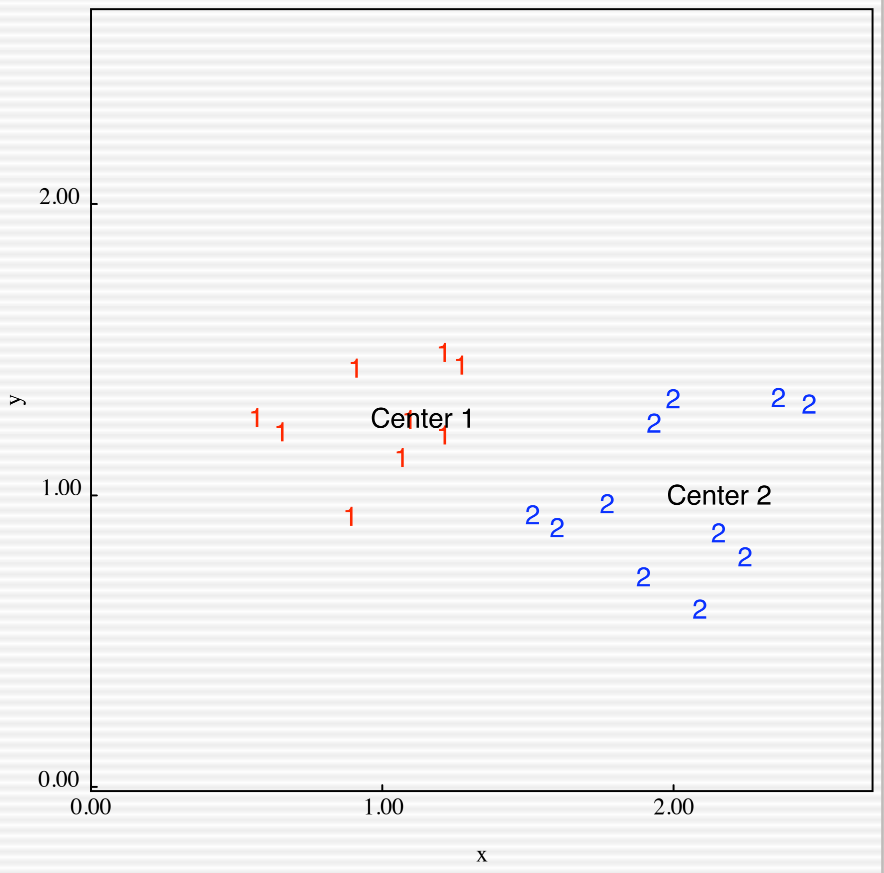
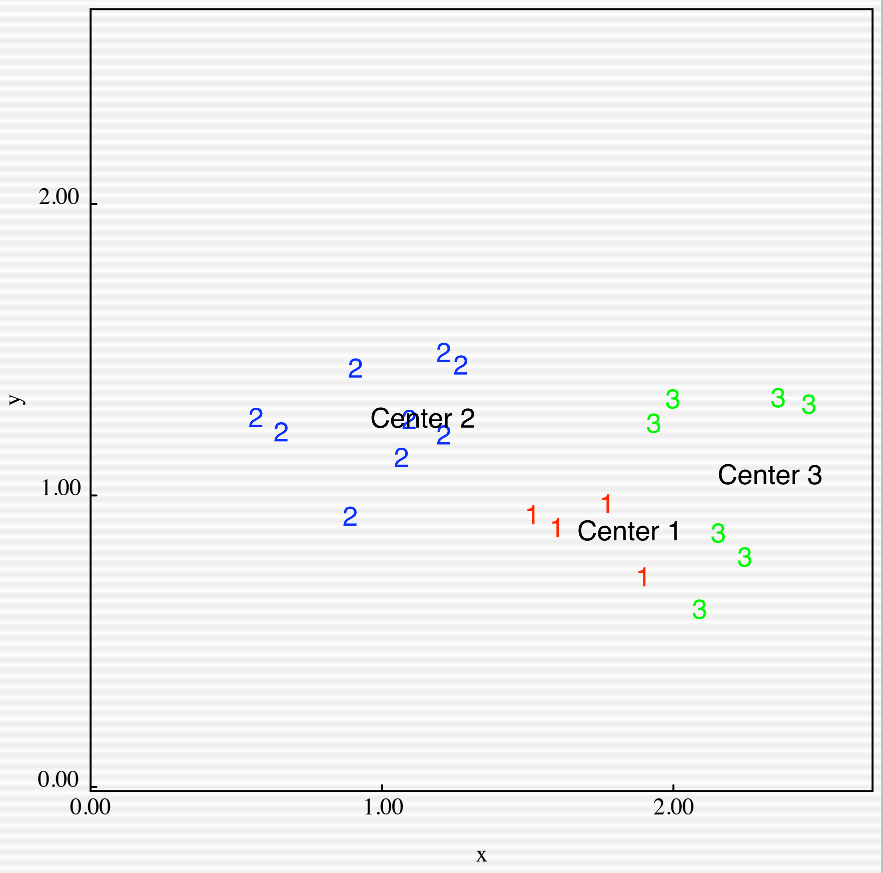
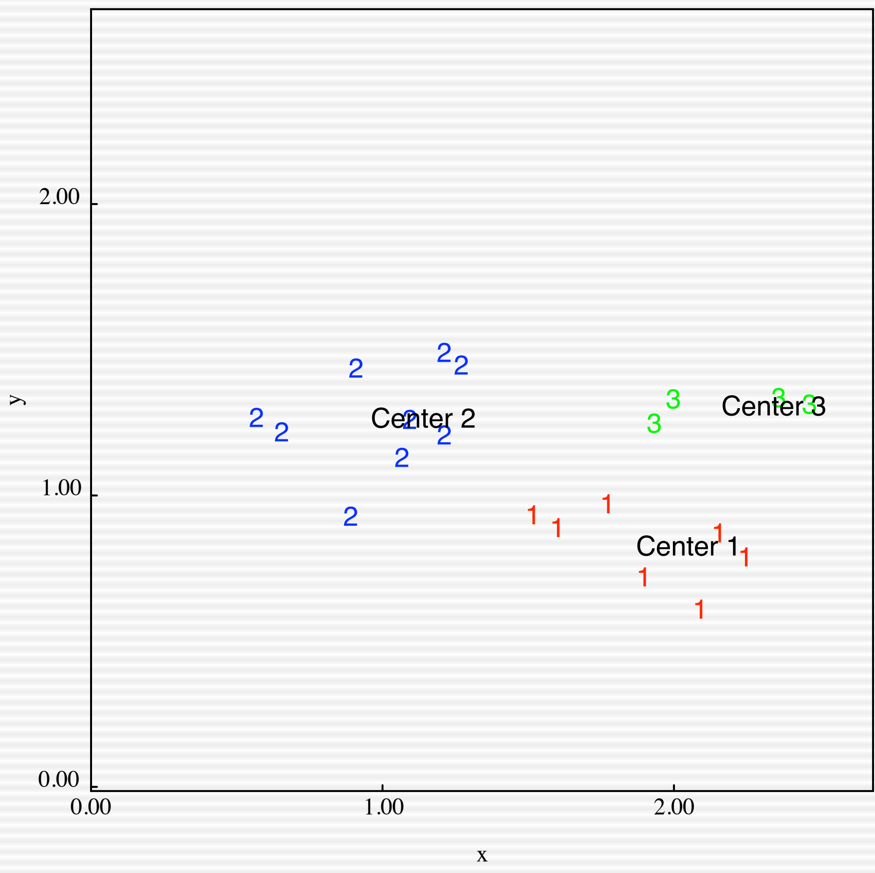


K-means execution example

Step 6

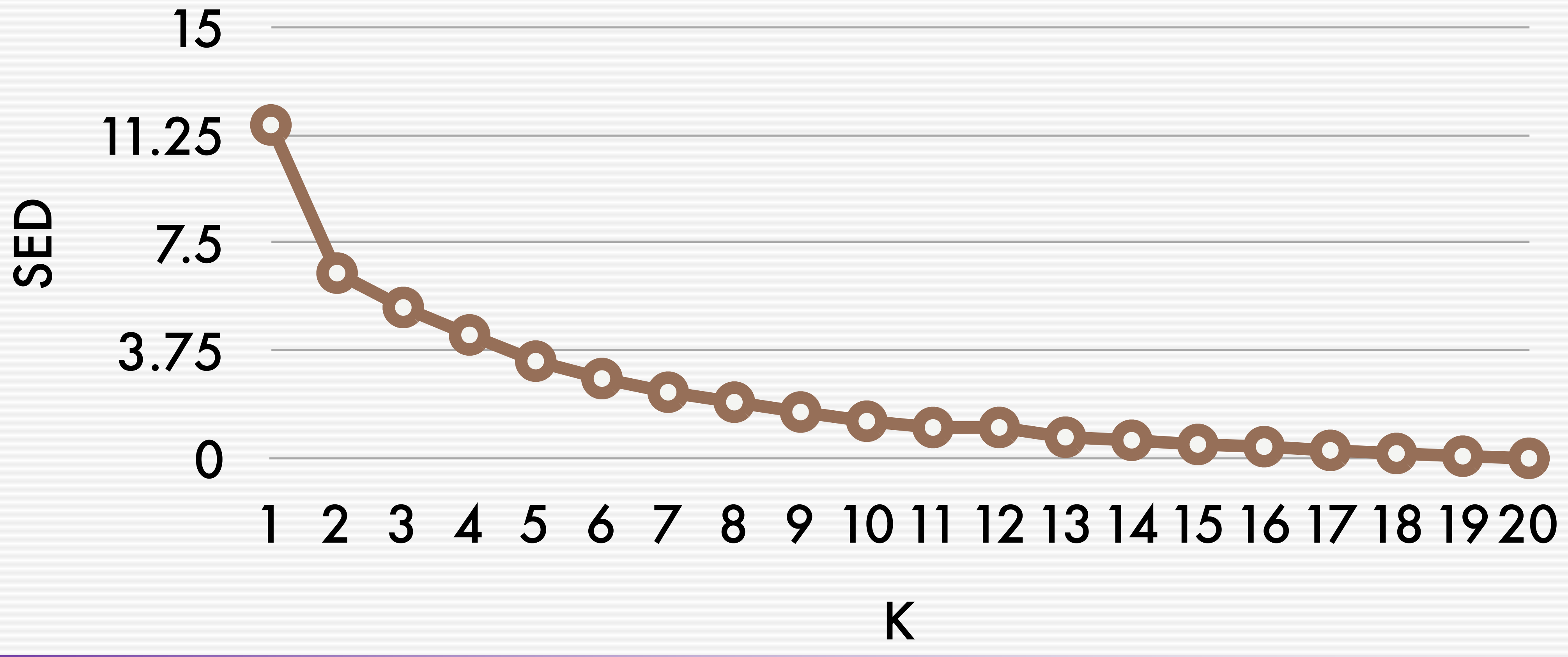


K-MEANS EXECUTION EXAMPLE

K=2	K=3	K=3 (different starting points)
		

How to choose K?

- The simplest approach is to start with K=1 and increase K until the squared error distortion (SED) stops decreasing
  - The problems is that K=N always achieves the value of **0** (each point is a cluster), so we always keep increasing K.
  - Generally, need to add further constraints (e.g. model complexity) to obtain non-trivial results

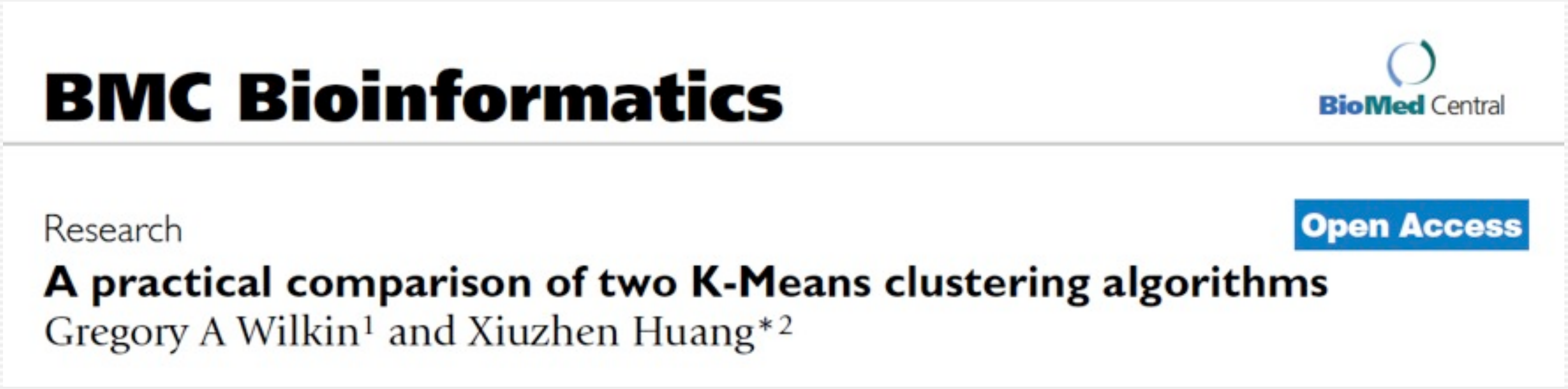


Conservative K-Means Algorithm

- Lloyd algorithm is fast but in each iteration it moves many data points, not necessarily causing better convergence.
- A more conservative method would be to move one point at a time only if it improves the overall **clustering cost**
  - The smaller the clustering cost of a partition of data points is the better that clustering is
  - Different methods (e.g. the squared error distortion) can be used to measure this clustering cost

K-MEANS "GREEDY" ALGORITHM

```
ProgressiveGreedyK-Means(k)
Select an arbitrary partition P into k clusters
while forever
   bestChange ← 0
   for every cluster C
      for every element i not in C
         if cost(P) – cost($P_{i \to C}$) > bestChange
            bestChange ← cost(P) – cost($P_{i \to C}$)
            $i^*$ ← I
            $C^*$ ←C
   if bestChange > 0
      Change partition P by moving $i^*$ to $C^*$
   else
     return P
```



BMC Bioinformatics

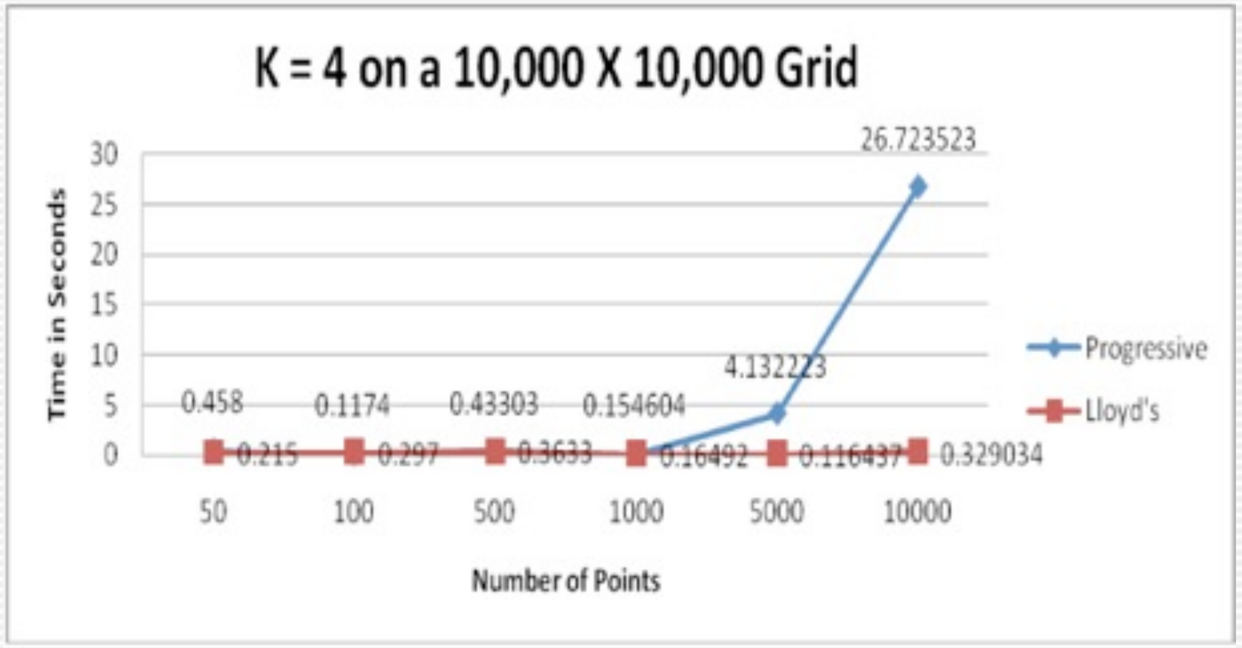
BioMed Central

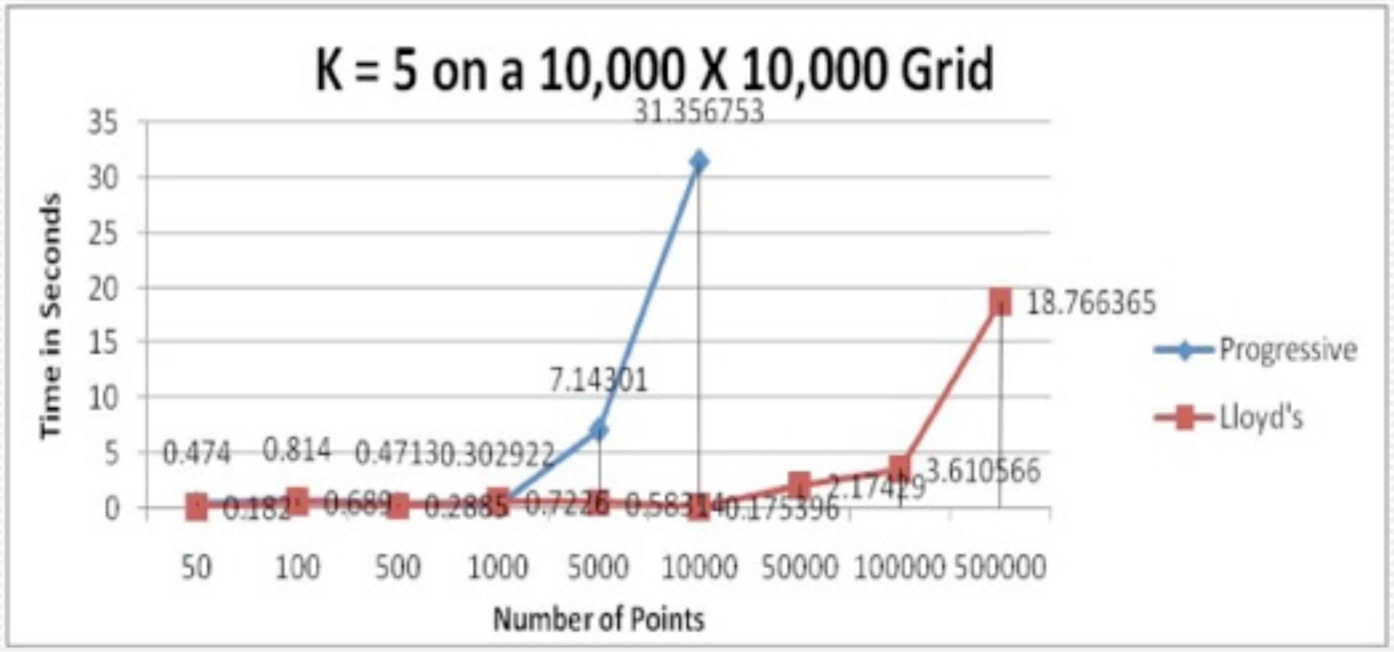
Research

Open Access

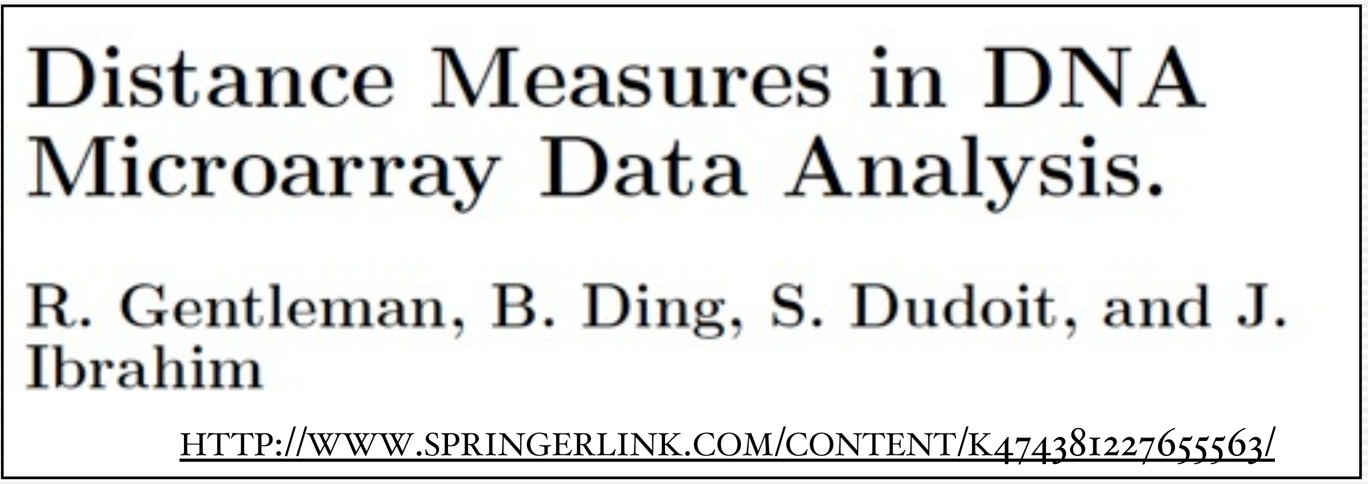
A practical comparison of two K-Means clustering algorithms

Gregory A Wilkin[1] and Xiuzhen Huang*[2]





Conclusion: Lloyd's in more efficient, both in run-time and in best found SED

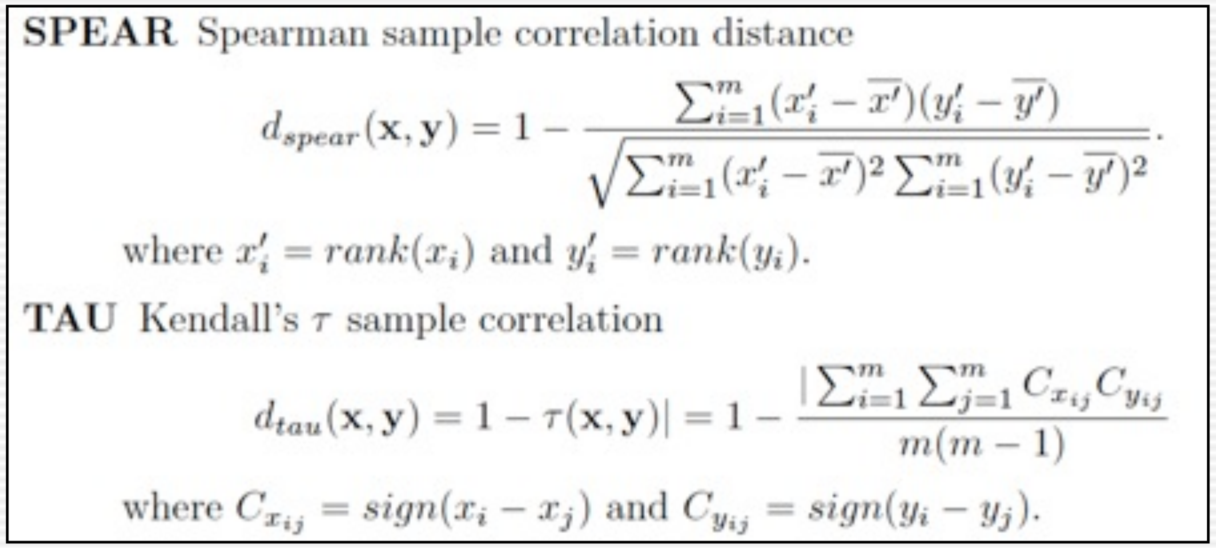


Distance Measures in DNA Microarray Data Analysis.

R. Gentleman, B. Ding, S. Dudoit, and J. Ibrahim

HTTP://WWW.SPRINGERLINK.COM/CONTENT/K474381227655563/

- Euclidean distance is not necessarily the best measure for co-expression.



SPEAR Spearman sample correlation distance

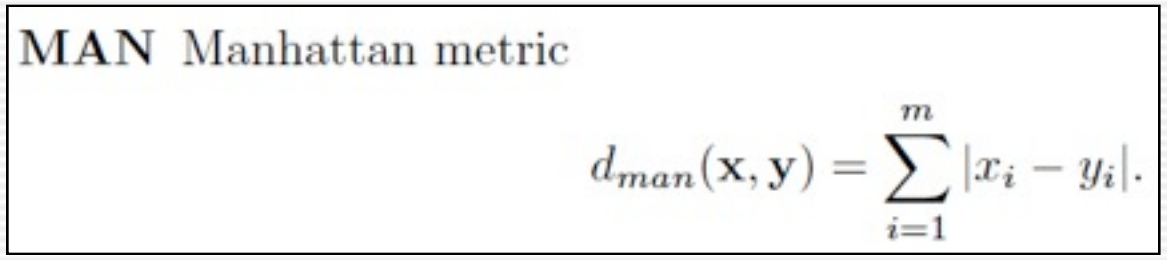
$$d_{spear}(\mathbf{x},\mathbf{y}) = 1 - \frac{\sum_{i=1}^{m}(x'_i - \overline{x'})(y'_i - \overline{y'})}{\sqrt{\sum_{i=1}^{m}(x'_i - \overline{x'})^2 \sum_{i=1}^{m}(y'_i - \overline{y'})^2}}.$$

where $x'_i = rank(x_i)$ and $y'_i = rank(y_i)$.

TAU Kendall's τ sample correlation

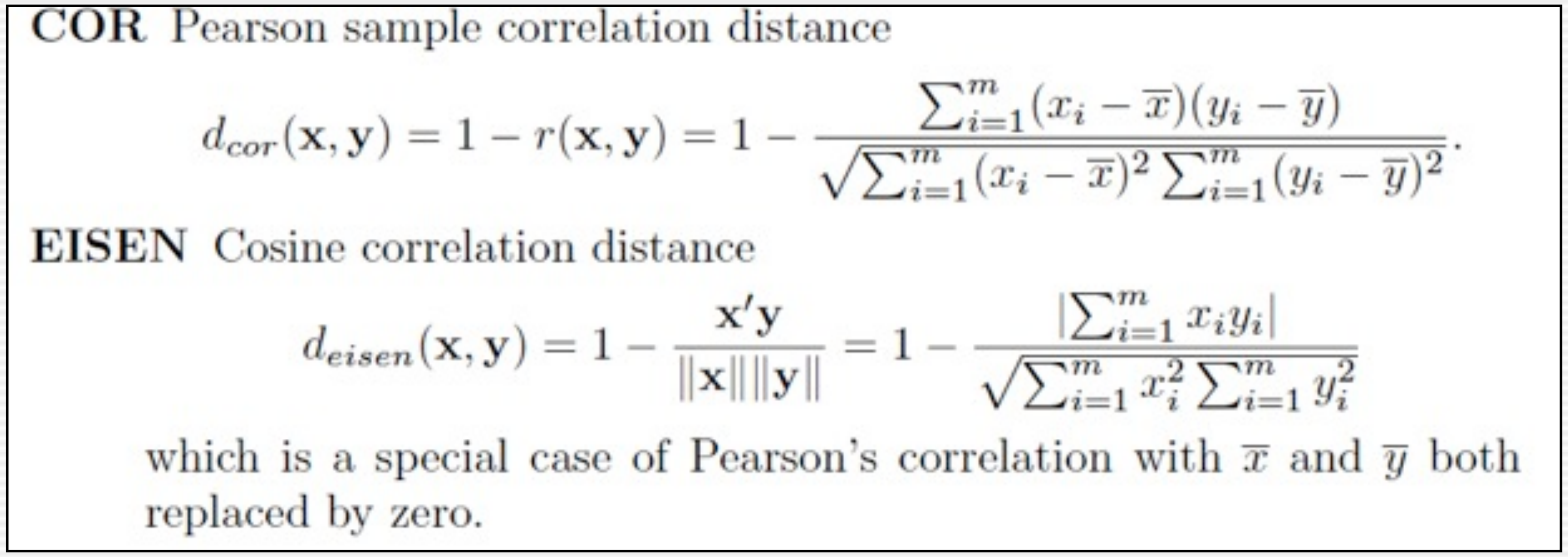
$$d_{tau}(\mathbf{x},\mathbf{y}) = 1 - \tau(\mathbf{x},\mathbf{y})| = 1 - \frac{|\sum_{i=1}^{m}\sum_{j=1}^{m} C_{x_{ij}} C_{y_{ij}}}{m(m-1)}$$

where $C_{x_{ij}} = sign(x_i - x_j)$ and $C_{y_{ij}} = sign(y_i - y_j)$.



MAN Manhattan metric

$$d_{man}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m} |x_i - y_i|.$$



COR Pearson sample correlation distance

$$d_{cor}(\mathbf{x},\mathbf{y}) = 1 - r(\mathbf{x},\mathbf{y}) = 1 - \frac{\sum_{i=1}^{m}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \overline{x})^2 \sum_{i=1}^{m}(y_i - \overline{y})^2}}.$$

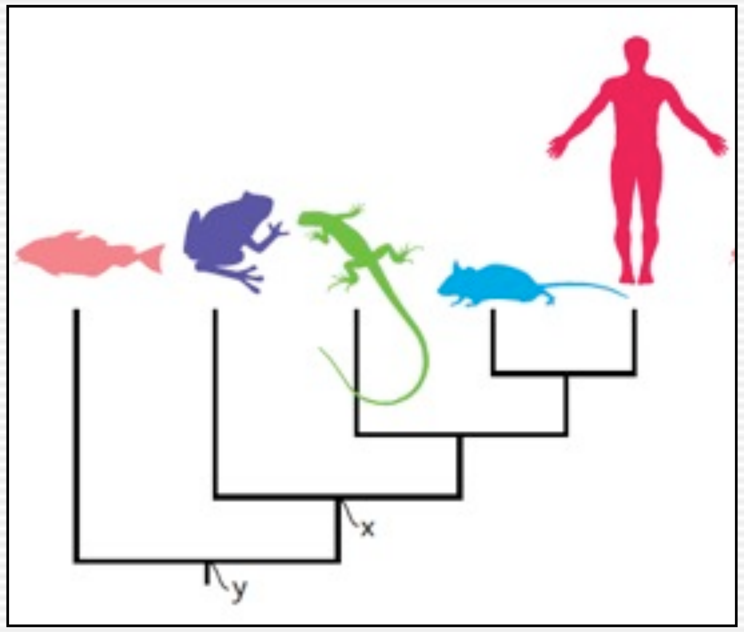
EISEN Cosine correlation distance

$$d_{eisen}(\mathbf{x},\mathbf{y}) = 1 - \frac{\mathbf{x}'\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = 1 - \frac{|\sum_{i=1}^{m} x_i y_i|}{\sqrt{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i^2}}$$

which is a special case of Pearson's correlation with $\overline{x}$ and $\overline{y}$ both replaced by zero.

Hierarchical clustering

- Instead of grouping into discrete clusters, produces a 'classification' tree, also called a dendrogram
- A more intuitive example is probably obtained from molecular sequence data (an early example of clustering applications)
- We have a collection of aligned nucleotide sequences from different species, and wish to construct their evolutionary hierarchy/history – a phylogeny.



HTTP://WWW.SCIENCEMAG.ORG/CGI/REPRINT/310/5750/979.PDF

HIERARCHICAL CLUSTERING

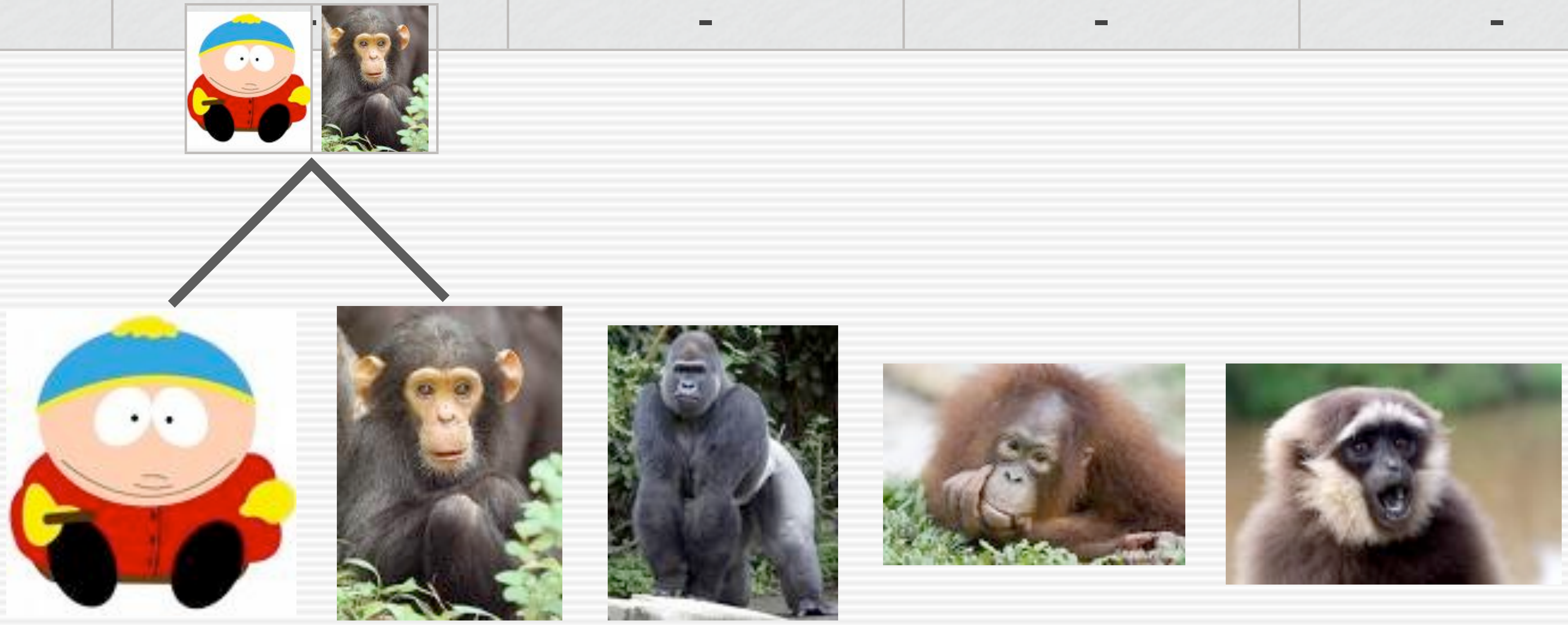
- Consider the following distance matrix on 5 nucleotide (partial mitochondrial genome) sequences. The values are *p-distances* defined as the number of nucleotide differences normalized by the length of the sequence.

	Human	Chimpanzee	Gorilla	Orangutan	Gibbon
Human	-	0.0882682	0.102793	0.159598	0.179688
Chimpanzee	-	-	0.106145	0.170759	0.1875
Gorilla	-	-	-	0.166295	0.1875
Orangutan	-	-	-	-	0.188616
Gibbon	-	-	-	-	-

Clustering procedure

- At each step, we select the two closest sequences and **join** them to form a clade.
- We then replace the two just joined sequences with their ancestor
- This reduces the size of the data matrix by one
- We need to compute the distances from the new ancestor to the remaining sequences

	Human	Chimpanzee	Gorilla	Orangutan	Gibbon
Human	-	**0.0882682**	0.102793	0.159598	0.179688
Chimpanzee	-	-	0.106145	0.170759	0.1875
Gorilla	-	-	-	0.166295	0.1875
Orangutan	-	-	-	-	0.188616
Gibbon	-	-	-	-	-



UPDATING DISTANCES

- There are multiple strategies for computing the distances to the new 'ancestral' sequence **a** that joins sequences **m** and **n**

Single Linkage	$d(x,a) = \min\left[d(x,m), d(x,n)\right]$
Complete Linkage	$d(x,a) = \max\left[d(x,m), d(x,n)\right]$
UPGMA Unweighted Pair Group Method with Arithmetic Mean	$d(x,a) = \frac{d(x,m)+d(x,n)}{2}$
WPGMA Weighted Pair Group Method with Arithmetic Mean	$d(x,a) = \frac{s(m)d(x,m)+s(n)d(x,n)}{s(m)+s(n)}$

s(**n**) counts the number of actual sequences represented by node **n**.

Example continued

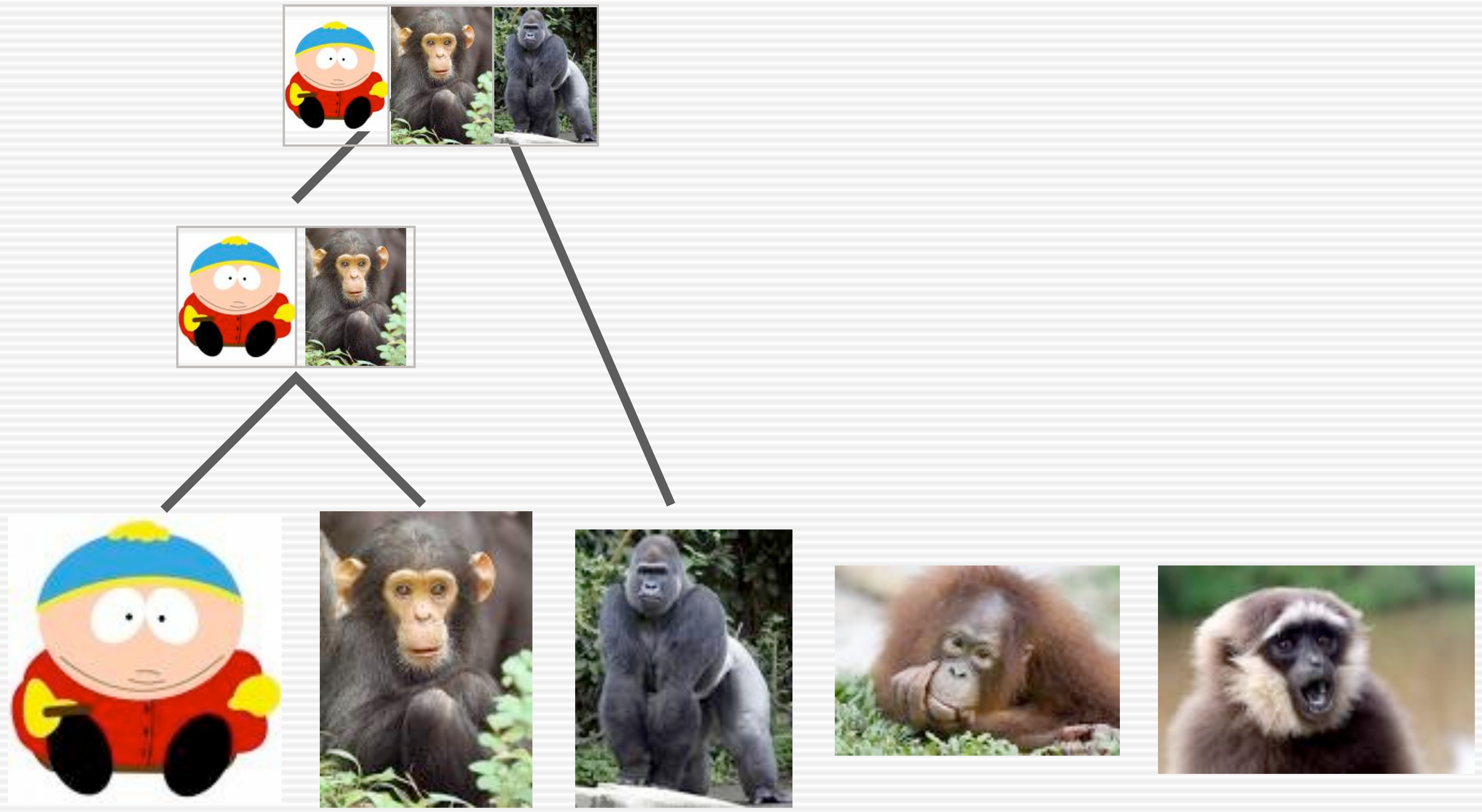
- Use complete linkage. Joining human and chimp...

	Human	Chimpanzee	Gorilla	Orangutan	Gibbon
Human	-	0.0882682	0.102793	0.159598	0.179688
Chimpanzee	-	-	0.106145	0.170759	0.1875
Gorilla	-	-	-	0.166295	0.1875
Orangutan	-	-	-	-	0.188616
Gibbon	-	-	-	-	-

	Human-Chimpanzee	Gorilla	Orangutan	Gibbon
Human-Chimpanzee	-	**0.106145**	**0.170759**	**0.1875**
Gorilla	-	-	0.166295	0.1875
Orangutan	-	-	-	0.188616
Gibbon	-	-	-	-

	Human-Chimpanzee	Gorilla	Orangutan	Gibbon
Human-Chimpanzee	-	0.106145	0.170759	0.1875
Gorilla	-	-	0.166295	0.1875
Orangutan	-	-	-	0.188616
Gibbon	-	-	-	-

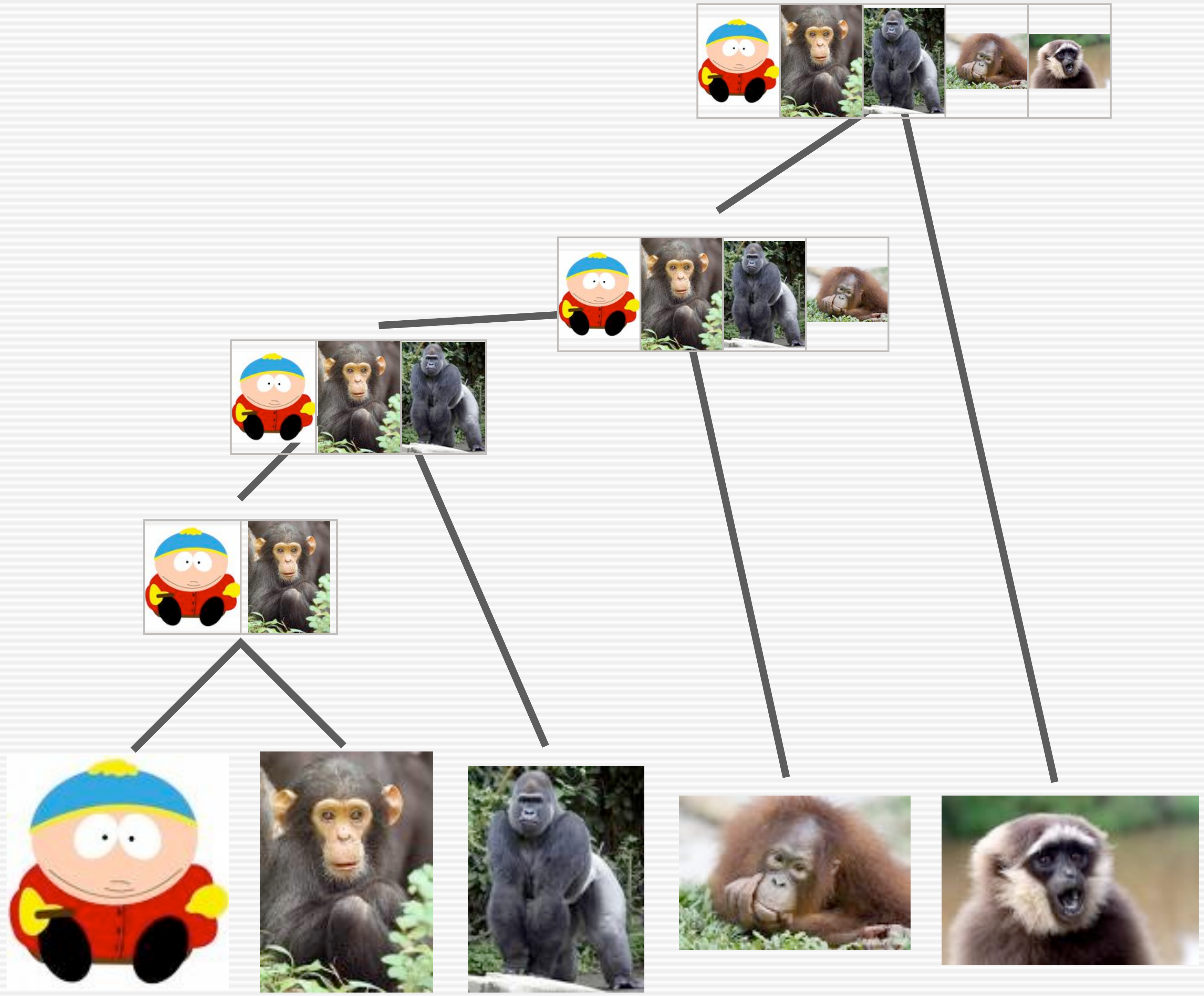
	Human-Chimpanzee-Gorilla	Orangutan	Gibbon
Human-Chimpanzee-Gorilla	-	0.170759	0.1875
Orangutan	-	-	0.188616
Gibbon	-	-	-



	Orangutan	Gibbon
Human-Chimpanzee-Gorilla	0.170759	0.1875
Orangutan	-	0.188616
Gibbon	-	-

$\Rightarrow$

	Gibbon
Hum-Chimp-Gor-Orang	0.188616
Gibbon	-



A note on WPGMA

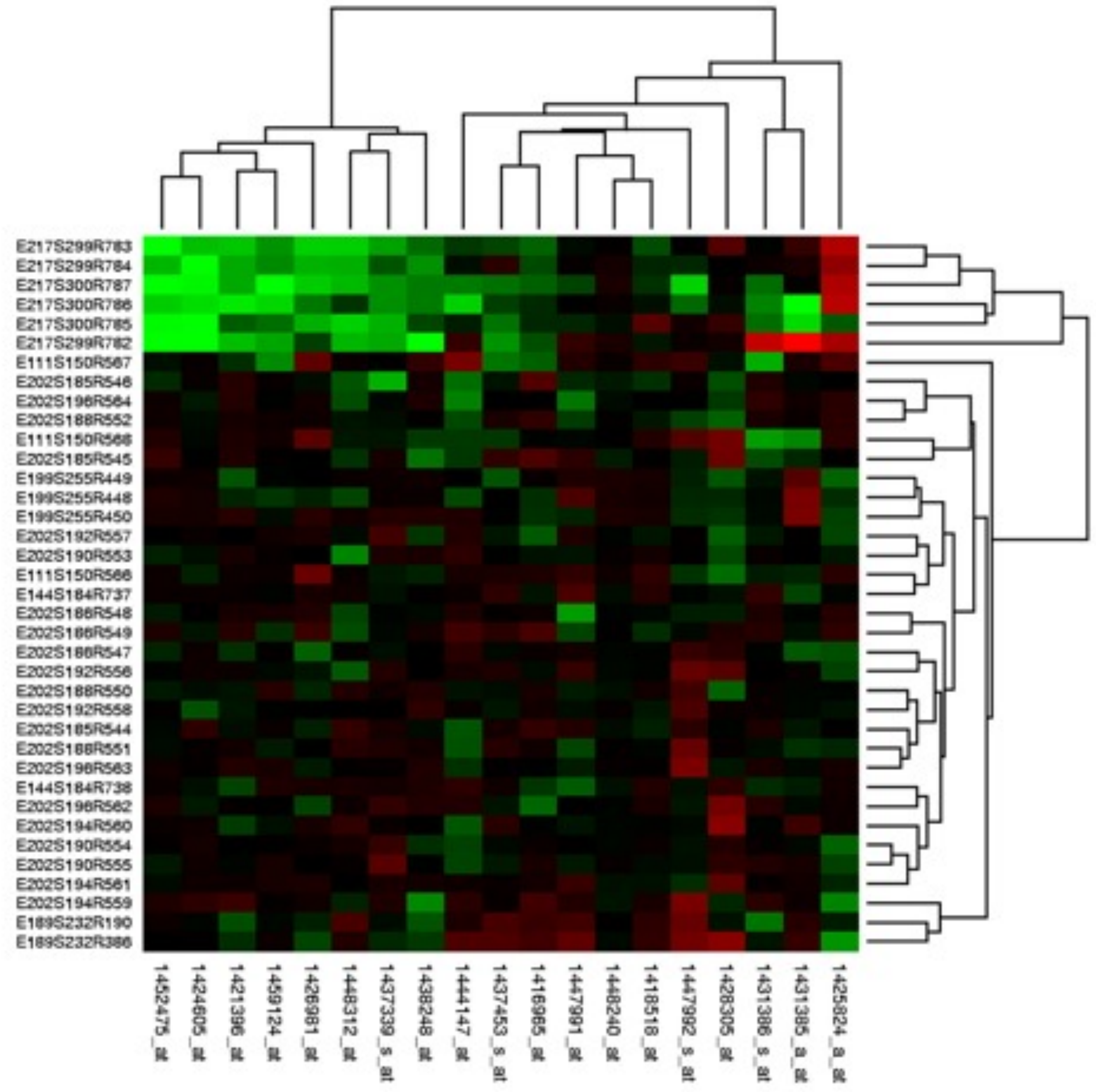
	Gorilla	Orangutan	Gibbon
Human-Chimpanzee	0.104469	0.165179	0.183594
Gorilla	-	0.166295	0.1875
Orangutan	-	-	0.188616
Gibbon	-	-	-

	Orangutan	Gibbon
Human-Chimpanzee-Gorilla	0.165551	0.184896
Orangutan	-	0.188616
Gibbon	-	-

- **d(HCG-Orang)** = 1/3 [2 d(HC-Orang) + d (Gor-Orang)]

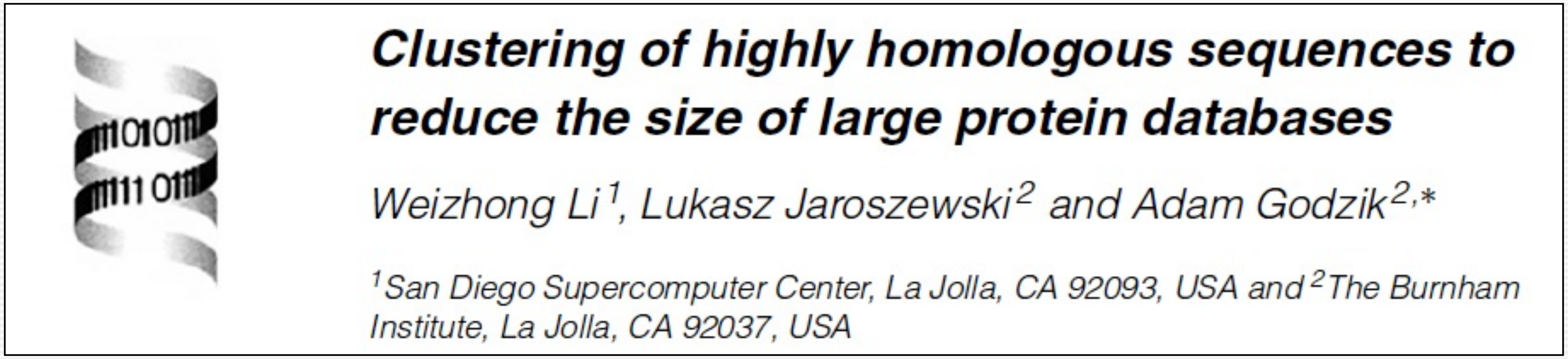
Back to microarrays...

- Clustering plots can be interpreted as gene/condition hierarchy



HTTP://UPLOAD.WIKIMEDIA.ORG/WIKIPEDIA/COMMONS/4/48/HEATMAP.PNG

A few other applications



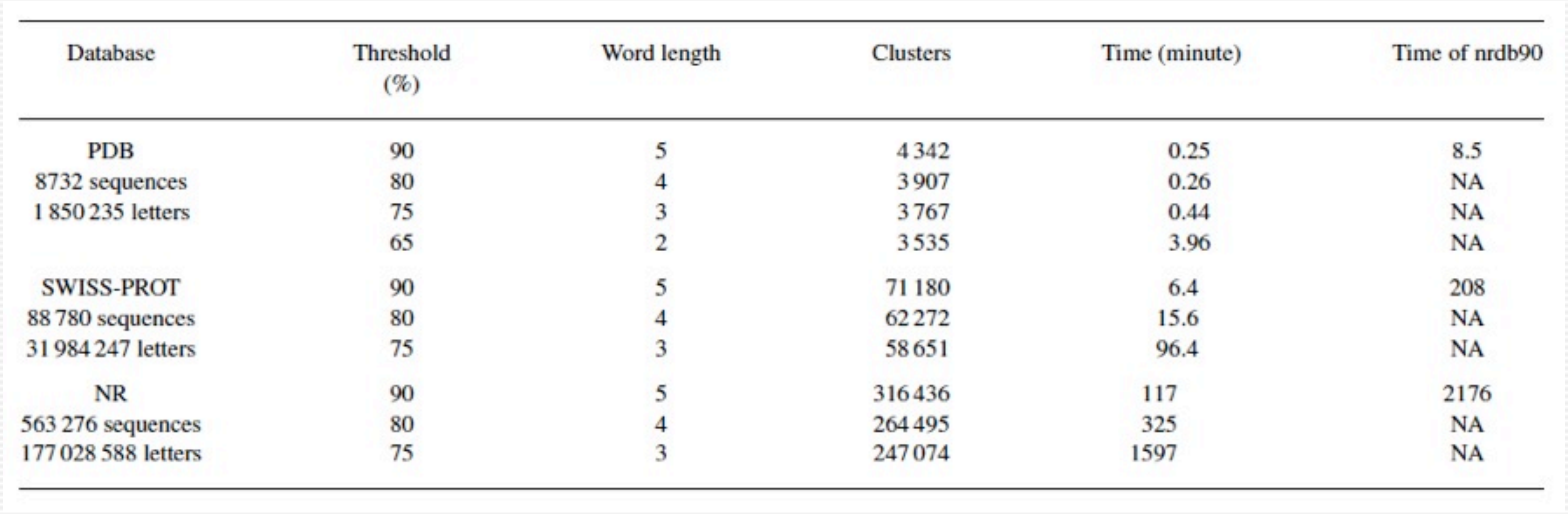
Clustering of highly homologous sequences to reduce the size of large protein databases

Weizhong Li[1], Lukasz Jaroszewski[2] and Adam Godzik[2,*]

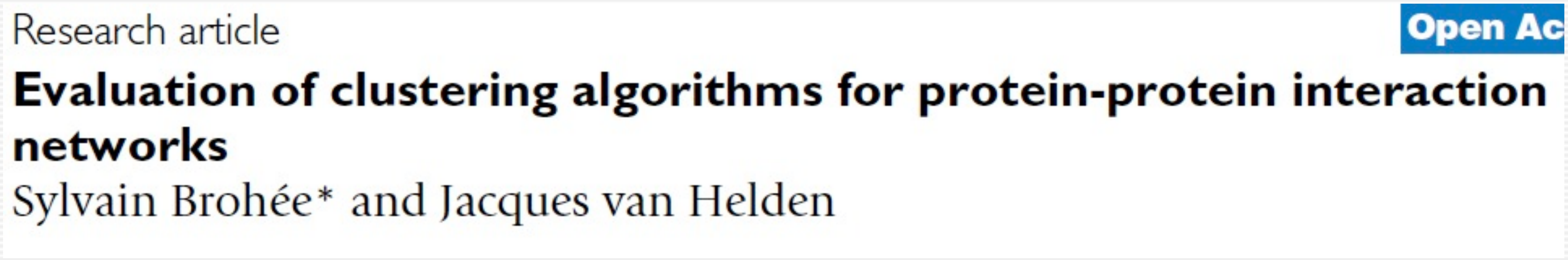
[1]San Diego Supercomputer Center, La Jolla, CA 92093, USA and [2]The Burnham Institute, La Jolla, CA 92037, USA

- Use clustering of similar sequences in protein databases to reduce complexity and speed up comparisons. Each cluster of similar sequences is represented by a single sequence.

- Complexity reduction is an important application of clustering



Database	Threshold (%)	Word length	Clusters	Time (minute)	Time of nrdb90
PDB	90	5	4 342	0.25	8.5
8732 sequences	80	4	3 907	0.26	NA
1 850 235 letters	75	3	3 767	0.44	NA
	65	2	3 535	3.96	NA
SWISS-PROT	90	5	71 180	6.4	208
88 780 sequences	80	4	62 272	15.6	NA
31 984 247 letters	75	3	58 651	96.4	NA
NR	90	5	316 436	117	2176
563 276 sequences	80	4	264 495	325	NA
177 028 588 letters	75	3	247 074	1597	NA



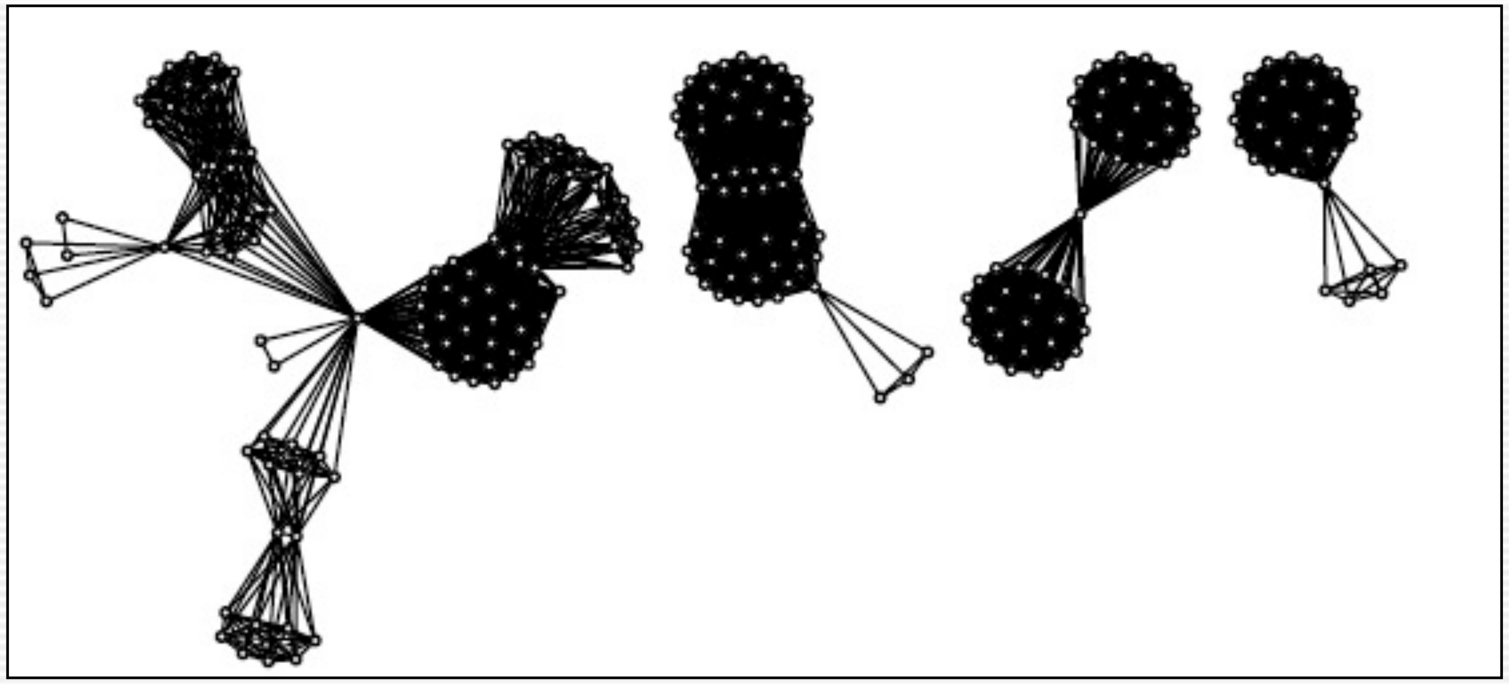
Research article

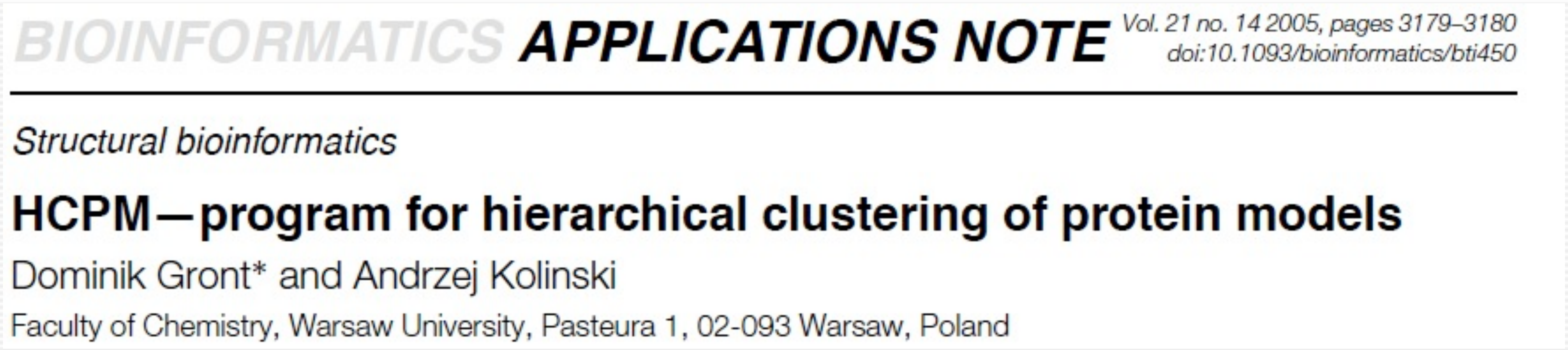
Open Ac

Evaluation of clustering algorithms for protein-protein interaction networks

Sylvain Brohée* and Jacques van Helden

- The structure of proteins interactions can be represented by a graph
  - Node = proteins, Edges = interactions
  - Look for clusters (densely connected components) in graphs





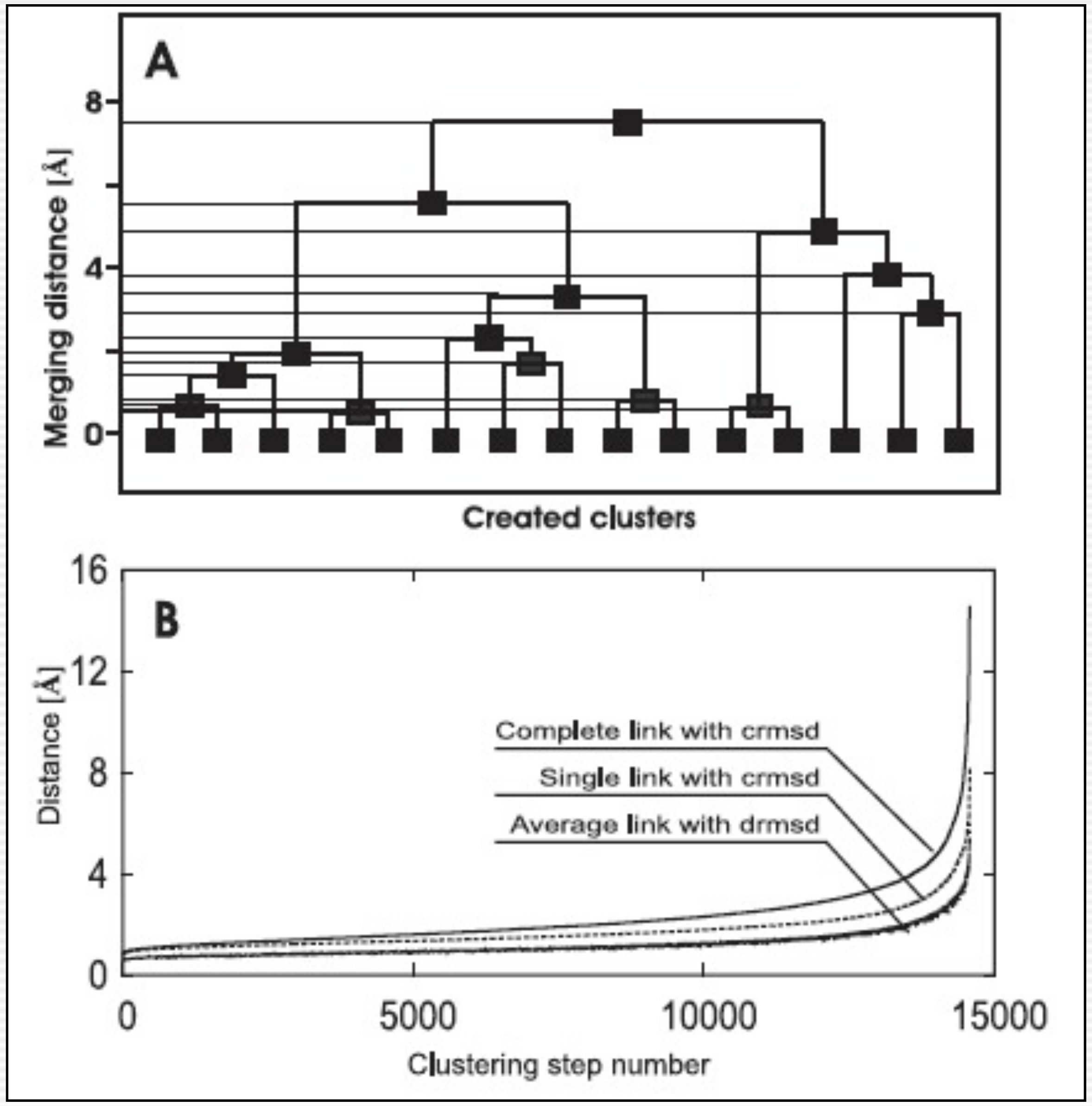
BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 14 2005, pages 3179–3180
doi:10.1093/bioinformatics/bti450

Structural bioinformatics

HCPM—program for hierarchical clustering of protein models

Dominik Gront* and Andrzej Kolinski
Faculty of Chemistry, Warsaw University, Pasteura 1, 02-093 Warsaw, Poland

- Hierarchical clustering to improve protein structure prediction by merging the predictions made by a large number of alternative conformation models



Further reading...

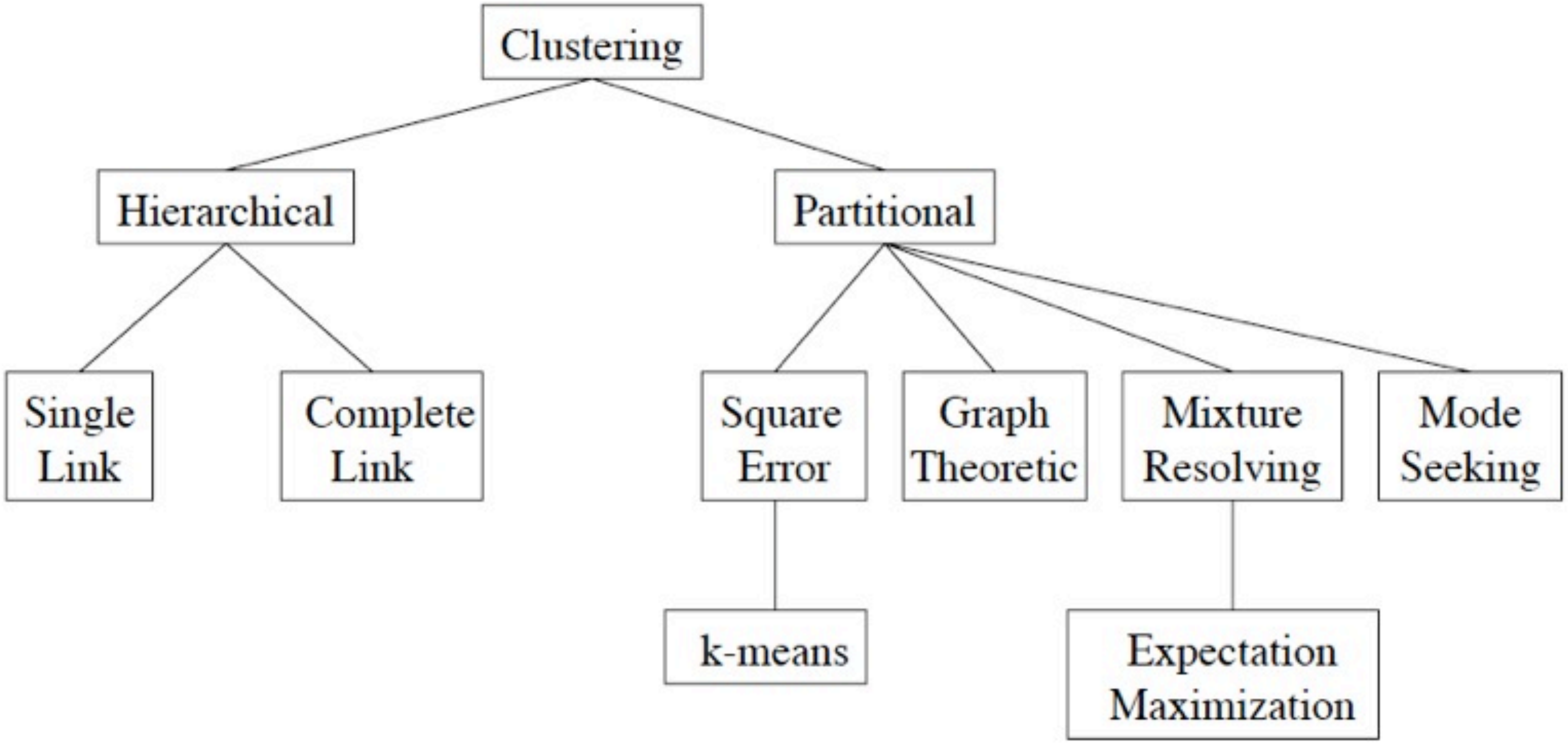


Figure 7. A taxonomy of clustering approaches.

Data Clustering: A Review

A.K. JAIN
Michigan State University
M.N. MURTY
Indian Institute of Science
AND
P.J. FLYNN
The Ohio State University

ACM Computing Surveys, Vol. 31, No. 3, September 1999

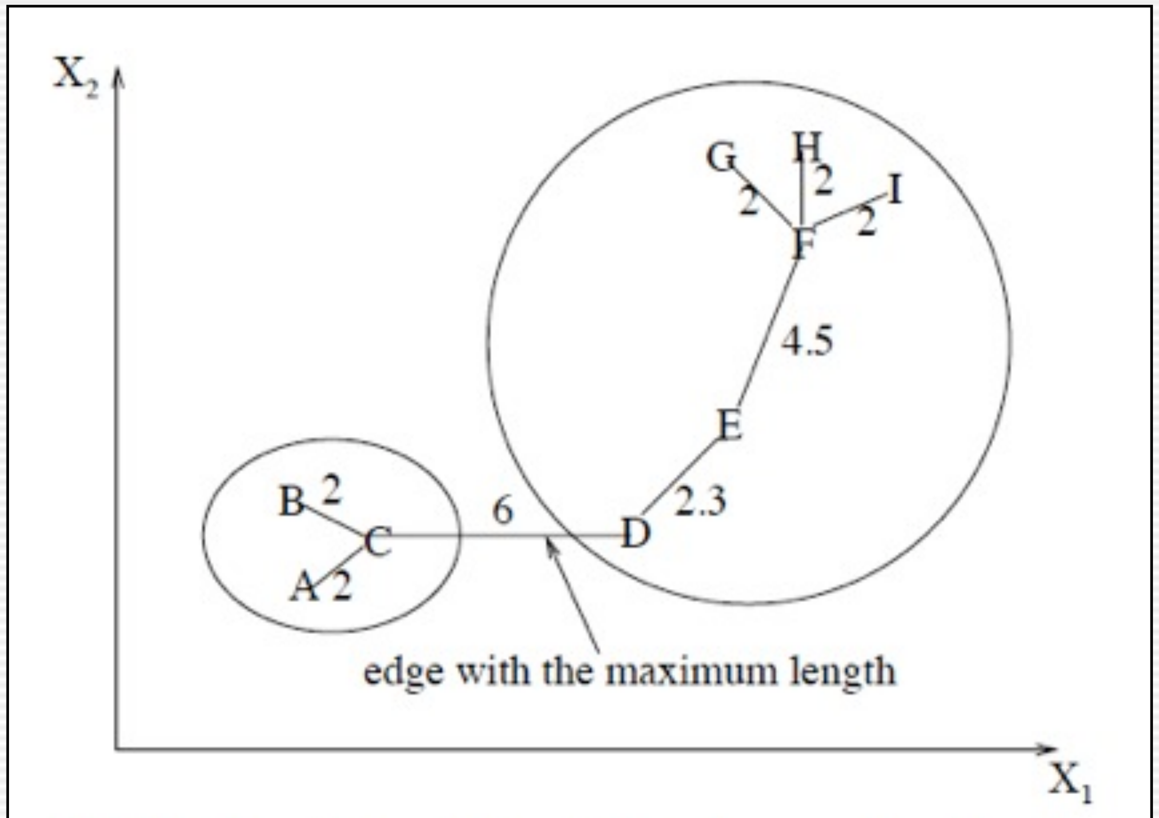


Figure 15. Using the minimal spanning tree to form clusters.

Defines the concept of 'element belongs to a partition with a probability'

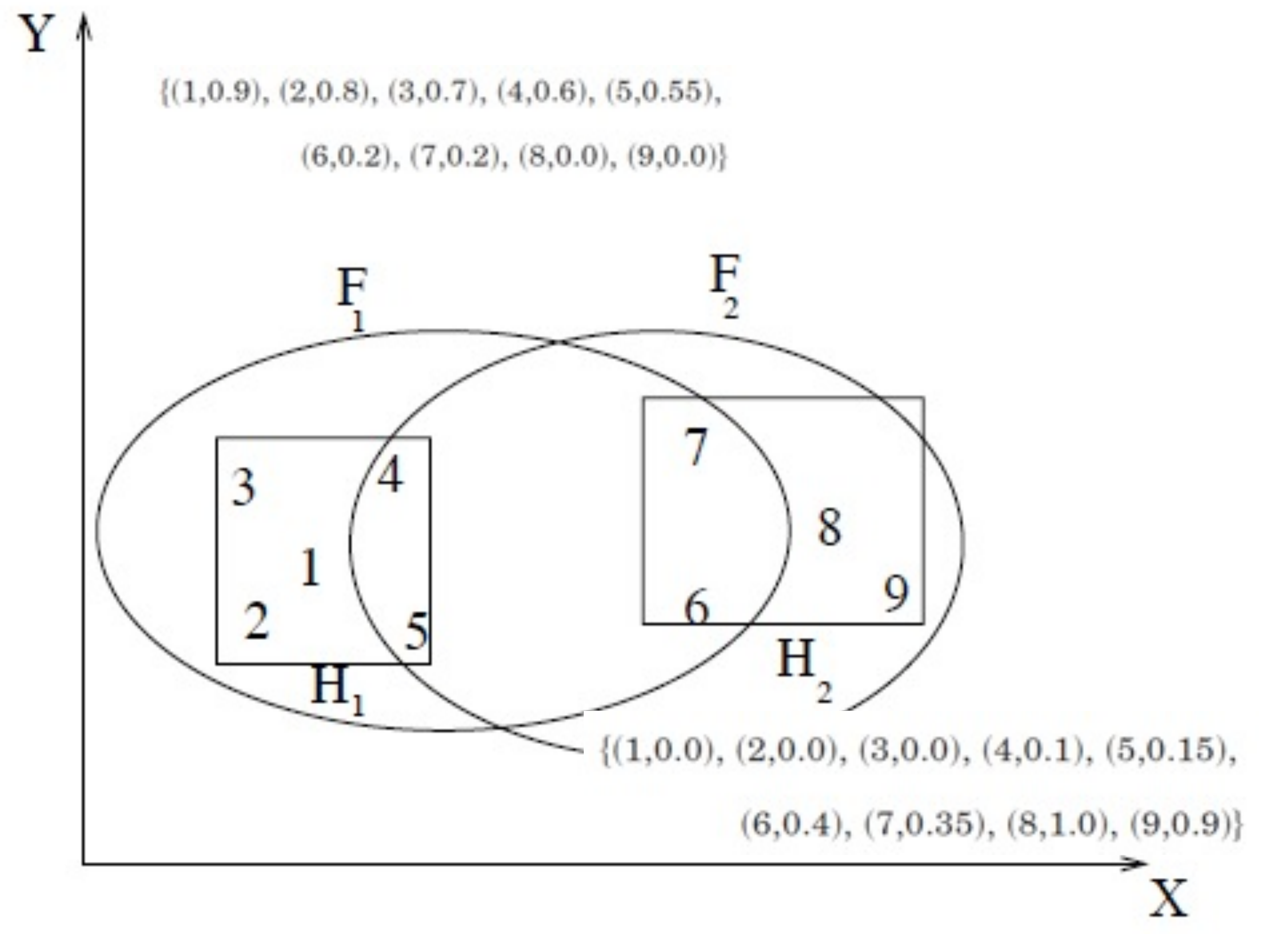


Figure 16. Fuzzy clusters.

Build a minimum spanning tree and delete longest edges to create partitions