Algorithms in Bioinformatics
Spring 2023
Lecture 4

Jason Ernst

University of California, Los Angeles

Administrative Announcements

- Paper 1 responses Tue 4/18
- HW2 – Chapter 9 Tue 4/18
- Discussion Friday – Chapter 9 and Project 1

Gene expression + Hierarchical clustering + K-means clustering

Lecture 4

April 13th, 2023

Some slides based on Andrew Moore and Manolis Kellis slides

Central Dogma

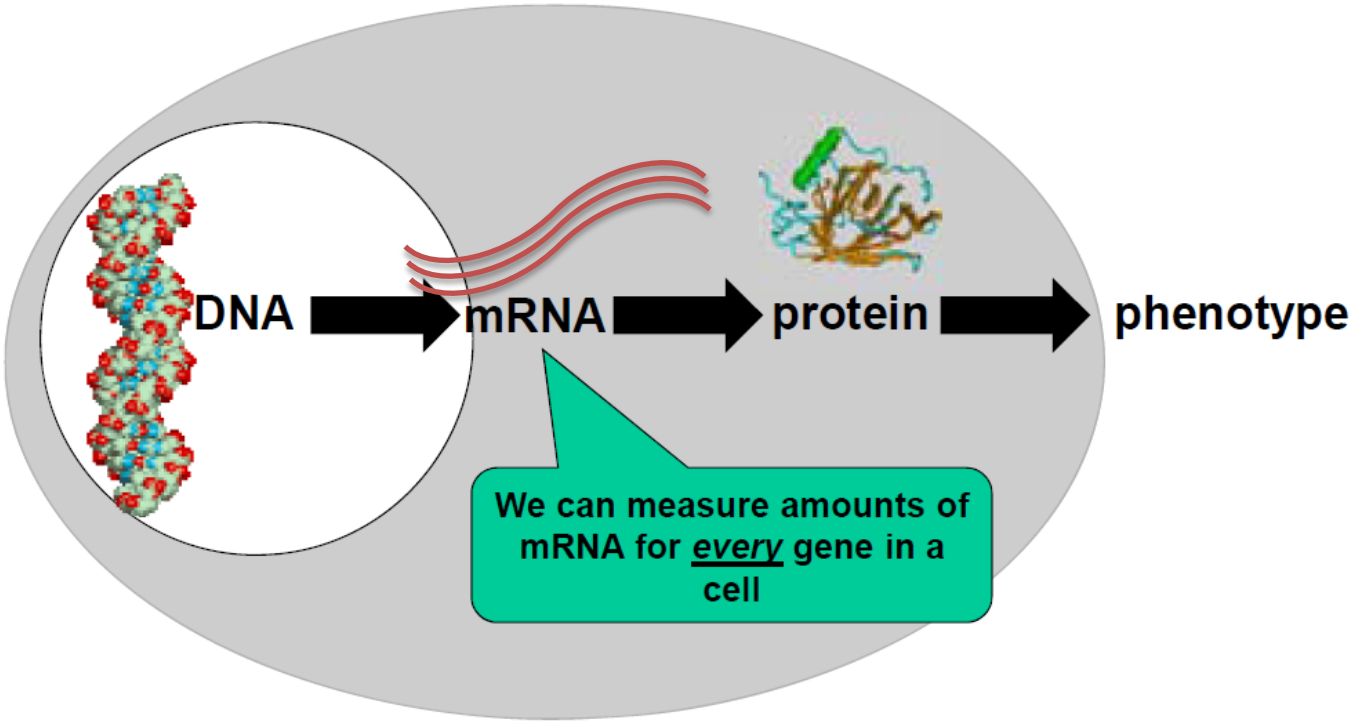
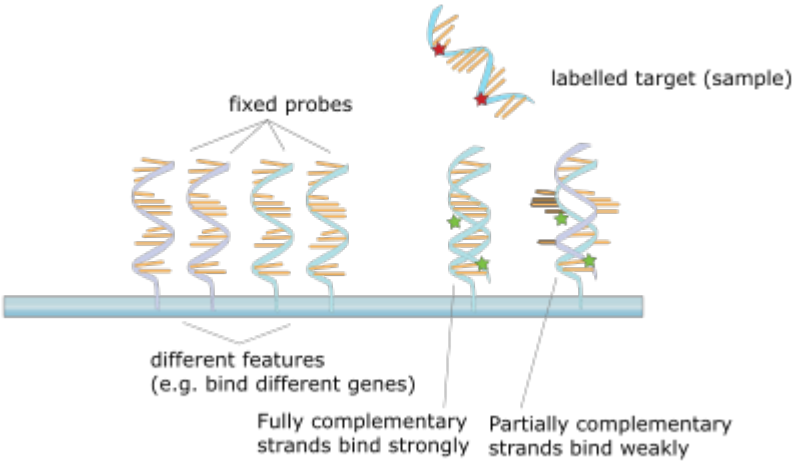


Image adapted from Manolis Kellis

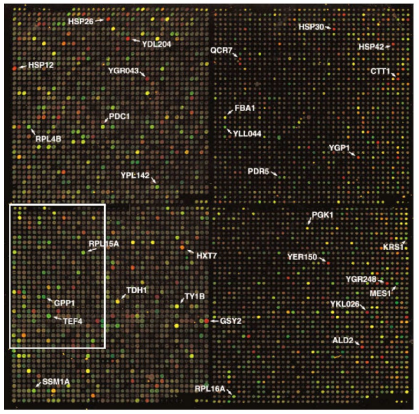
Measuring Gene Expression

- Organisms typically have on the order of thousands or tens of thousands of protein coding genes (e.g. ~20,000 genes in human)
- Gene expression profiling used to be limited to a single gene at a time. This changed with the publication of the first microarray in 1995.
- By 1997, scaled up to measure expression of all yeast genes



Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale

Joseph L. DeRisi, Vishwanath R. Iyer, Patrick O. Brown*



SCIENCE • VOL. 278 • 24 OCTOBER 1997

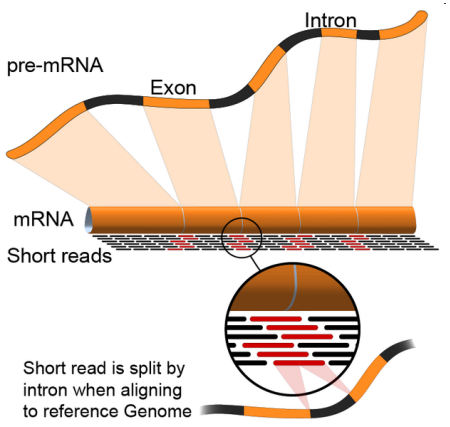
(Data from chapter 8)

Image source: wikipedia

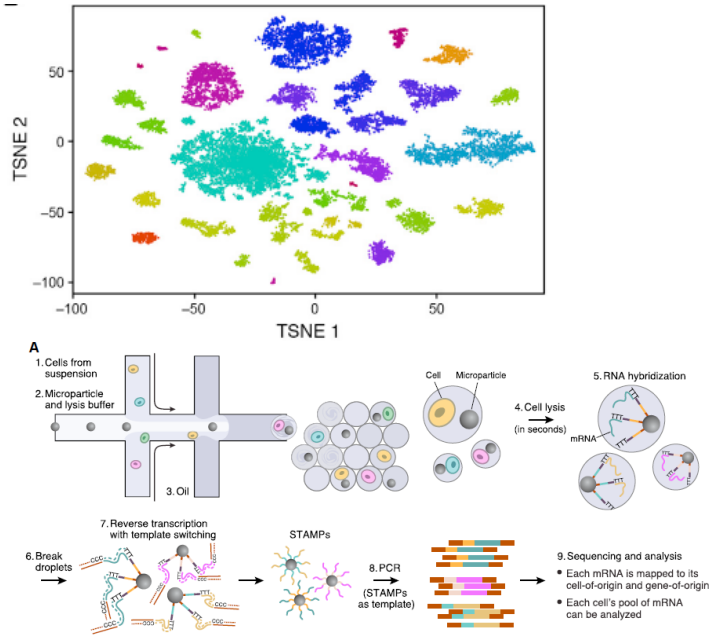
Measuring Gene Expression

Two additional major waves of technologies for measuring gene expression

RNA-seq – bulk samples



Single cell RNA-seq



Macosko et al, *Cell* 2015

Computational problem of clustering genes remains common across technologies
Additional computational challenges associated with each technology –
Discussed in CM121/CM221

Image source: wikipedia

Gene Expression Data

Gene Symbol	0h	0.5h	3h	6h	12h
ZFX	-0.027	0.158	0.169	0.193	-0.165
ZNF133	0.183	-0.068	-0.134	-0.252	0.177
USP2	-0.67	-0.709	-0.347	-0.779	-0.403
DSCR1L1	-0.923	-0.51	-0.718	-0.512	-0.668
WNT5A	-0.471	-0.264	-0.269	-0.154	-0.254
VHL	-0.327	-0.378	-0.229	-0.264	-0.072
TCF3	-0.021	0.129	-0.209	-0.245	0.036
TCN2	-0.492	-0.41	-0.306	-0.494	-0.273
TIMP1	-0.111	0.351	0.168	0.129	-0.293
SERPINA7	-0.468	-0.488	-0.199	-0.144	-0.185
THBD	-1.013	-0.895	-0.743	-0.601	-0.543
EPHA2	0.13	0.313	0.645	-0.155	0.28
RBM5	0.015	-0.139	-0.14	-0.432	0.303
SFRS10	0.314	0.235	0.313	0.482	-0.303
SLC16A4	0.097	-0.432	-0.294	0.17	0.853
C20orf16	-0.203	0.147	0.267	0.29	0.508
RBM3	-0.253	0.987	0.451	0.245	-0.313
C3AR1	-0.364	0.109	-0.063	-0.129	-0.415
MLF2	-0.193	0.168	-0.005	-0.067	-0.06
ABCC5	0.161	0.025	-0.156	0.097	0.272
DAB2	-0.09	-0.079	-0.56	-1.054	-0.933
POLRMT	-0.1	0.032	-0.344	-0.307	-0.197
DECR1	0.191	-0.281	-0.242	0.103	0.005

- Rows – genes
- Columns – different experimental conditions

Gene Expression Data

- Genes with similar gene expression patterns across experimental conditions often involved in same biological process or co-regulated (e.g. bound by same transcription factor)

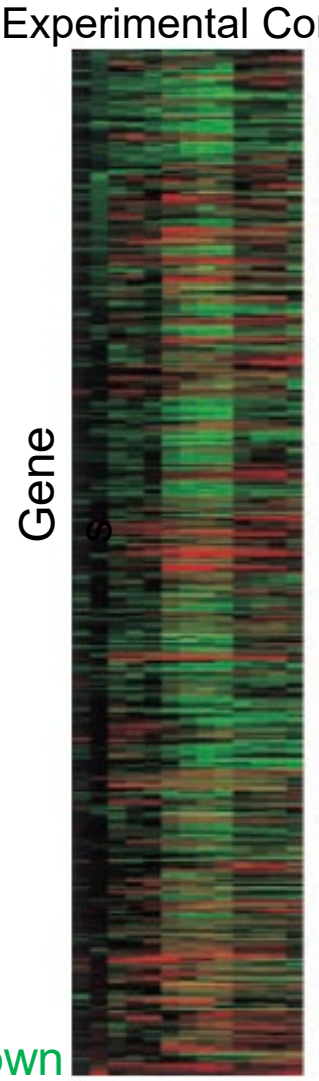


Image from *Eisen et al,* 1998

Gene Expression Data

- Genes with similar gene expression patterns across experimental conditions often involved in same biological process or co-regulated (e.g. bound by same transcription factor)

- By identifying sets of genes with similar expression patterns can lead to insight into biological processes associated with the conditions, gene regulatory mechanism, and roles of genes with unknown function

    - Example: identify sequence patterns around transcription start sites of genes with similar expression patterns (ch. 2)

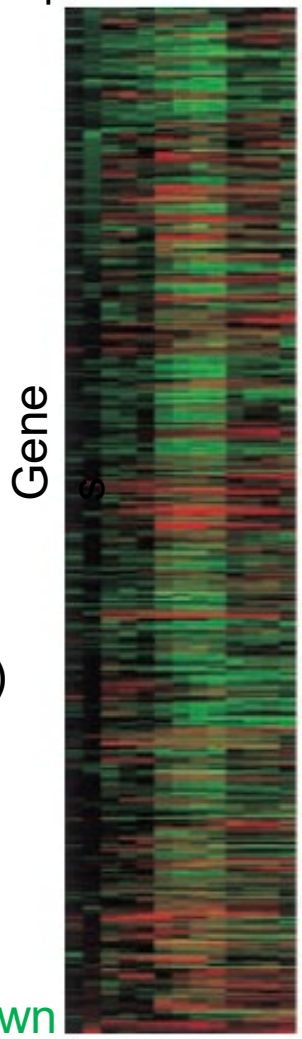


Image from *Eisen et al,* 1998

Gene Expression Data

- Genes with similar gene expression patterns across experimental conditions often involved in same biological process or co-regulated (e.g. bound by same transcription factor)

- By identifying sets of genes with similar expression patterns can lead to insight into biological processes associated with the conditions, gene regulatory mechanism, and roles of genes with unknown function

    - Example: identify sequence patterns around transcription start sites of genes with similar expression patterns (ch. 2)

How should we identify sets of genes with similar expression patterns?

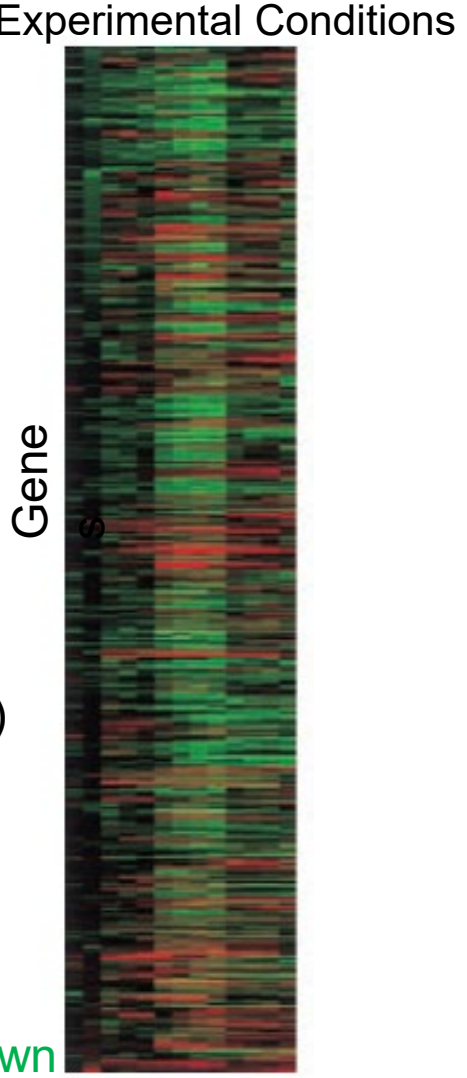
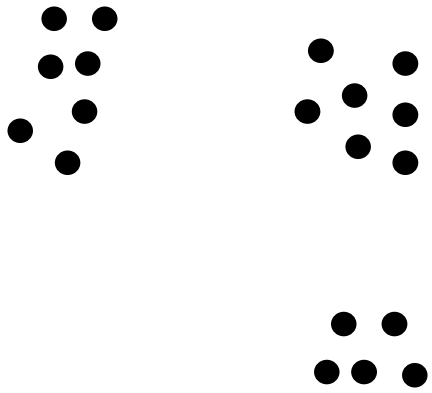


Image from *Eisen et al,* 1998

Clustering

- Group unlabeled data points
- Informally:
  - Data points “near” each other in the same clusters
  - Data points “far” from each other in different clusters



Hierarchical clustering

Lecture 4

April 13th, 2023

Some slides based on Andrew Moore and Manolis Kellis slides

Hierarchical Clustering

Proc. Natl. Acad. Sci. USA
Vol. 95, pp. 14863–14868, December 1998
Genetics

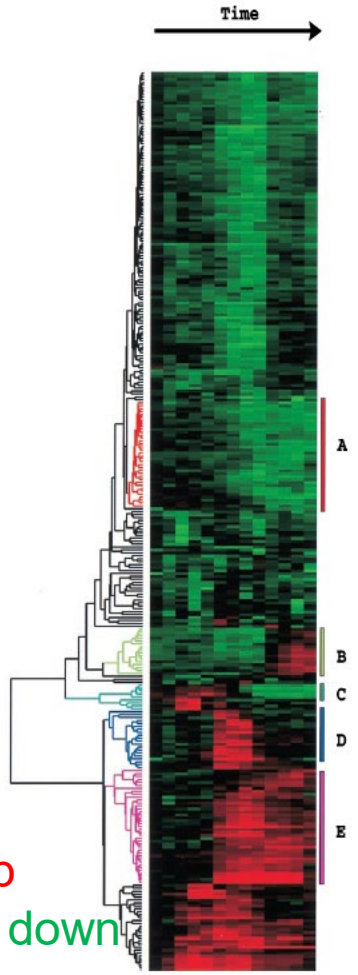
Cluster analysis and display of genome-wide expression patterns

MICHAEL B. EISEN*, PAUL T. SPELLMAN*, PATRICK O. BROWN†, AND DAVID BOTSTEIN*‡

*Department of Genetics and †Department of Biochemistry and Howard Hughes Medical Institute, Stanford University School of Medicine, 300 Pasteur Avenue, Stanford, CA 94305

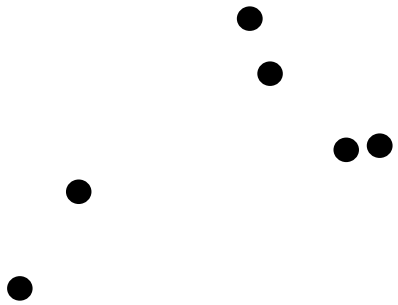
Contributed by David Botstein, October 13, 1998

Cited >20,000 times (google scholar)



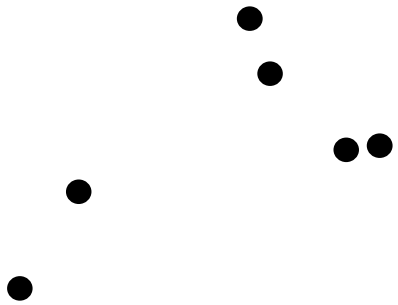
Red – up
Green – down

Hierarchical Clustering



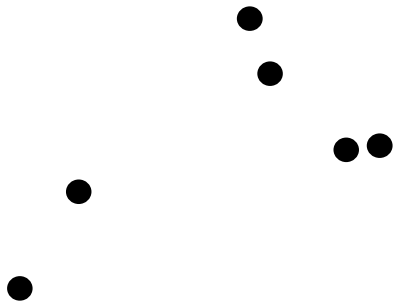
Hierarchical Clustering

1. Initially each point is its own cluster



Hierarchical Clustering

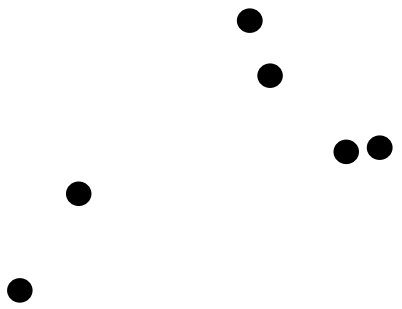
1. Initially each point is its own cluster
2. Find pair of clusters with smallest distance between them or equivalently are the most similar



Hierarchical Clustering

1. Initially each point is its own cluster
2. Find pair of clusters with smallest distance between them or equivalently are the most similar

How should we determine distance between pairs of data points?



Hierarchical Clustering Distance Measures

Common approaches to measure distance:

Table 1 Gene expression similarity measures

Manhattan distance (city-block distance, L1 norm)	$d_{fg} = \sum_c \lvert e_{fc} - e_{gc} \rvert$
Euclidean distance (L2 norm)	$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2}$
Mahalanobis distance	$d_{fg} = (e_f - e_g)' \Sigma^{-1} (e_f - e_g)$, where Σ is the (full or within-cluster) covariance matrix of the data
Pearson correlation (centered correlation)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}}$
Uncentered correlation (angular separation, cosine angle)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c e_{fc} e_{gc}}{\sqrt{\sum_c e_{fc}^2 \sum_c e_{gc}^2}}$
Spearman rank correlation	As Pearson correlation, but replace e_{gc} with the rank of e_{gc} within the expression values of gene g across all conditions $c = 1 \ldots C$
Absolute or squared correlation	$d_{fg} = 1 - \lvert r_{fg} \rvert$ or $d_{fg} = 1 - r_{fg}^2$

d_{fg}, distance between expression patterns for genes f and g. e_{gc}, expression level of gene g under condition c.

D'haeseleer, Nature Biotech 2005

Hierarchical Clustering Distance Measures

Pearson correlation (centered correlation)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}}$

d_{fg}, distance between expression patterns for genes f and g. e_{gc}, expression level of gene g under condition c.

Pearson correlation is popular since it clusters based on shape and relative changes which can often be more informative than absolute expression levels

Question: How could pearson correlation fail to provide output?

Hierarchical Clustering Distance Measures

Pearson correlation (centered correlation)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}}$

d_{fg}, distance between expression patterns for genes f and g. e_{gc}, expression level of gene g under condition c.

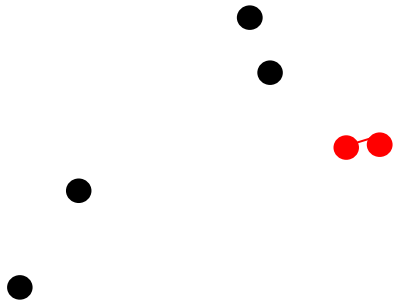
Pearson correlation is popular since it clusters based on shape and relative changes which can often be more informative than absolute expression levels

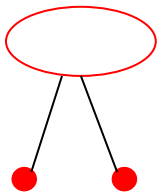
Question: How could pearson correlation fail to provide output?

Undefined if no variance. Typically filter genes that do not change expression before clustering.

Hierarchical Clustering

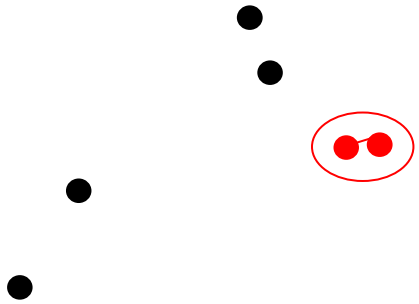
1. Initially each point is its own cluster
2. Find pair of clusters with smallest distance between them or equivalently are the most similar

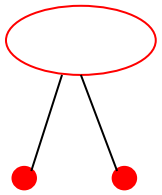




Hierarchical Clustering

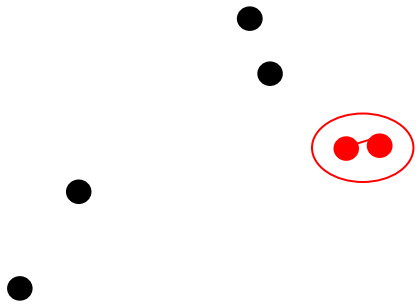
1. Initially each point is its own cluster
2. Find pair of clusters with smallest distance between them or equivalently are the most similar
3. Merge into parent cluster



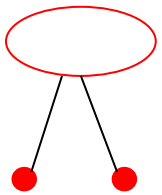


Hierarchical Clustering

1. Initially each point is its own cluster
2. Find pair of clusters with smallest distance between them or equivalently are the most similar
3. Merge into parent cluster
4. Repeat



Question: How should we determine distance between two clusters when one or both clusters contains more than one point?



Measuring Distance between Clusters

- Single Linkage Clustering

$$\mathrm{D}(\mathrm{X},\mathrm{Y}) = min_{x\in X, y\in Y} d(x,y)$$

- Complete Linkage Clustering

$$\mathrm{D}(\mathrm{X},\mathrm{Y}) = max_{x\in X, y\in Y} d(x,y)$$

- Average Linkage Clustering

$$\mathrm{D}(\mathrm{X},\mathrm{Y}) = \frac{1}{|X||Y|}\sum_{x\in X}\sum_{y\in Y} d(x,y)$$

- Centroid Linkage Clustering

$$\mathrm{D}(\mathrm{X},\mathrm{Y}) = ||c_X - c_Y||$$

where c_X and c_Y are the mean of *X* and *Y* and data assumed to be in $\mathbb{R}^d$

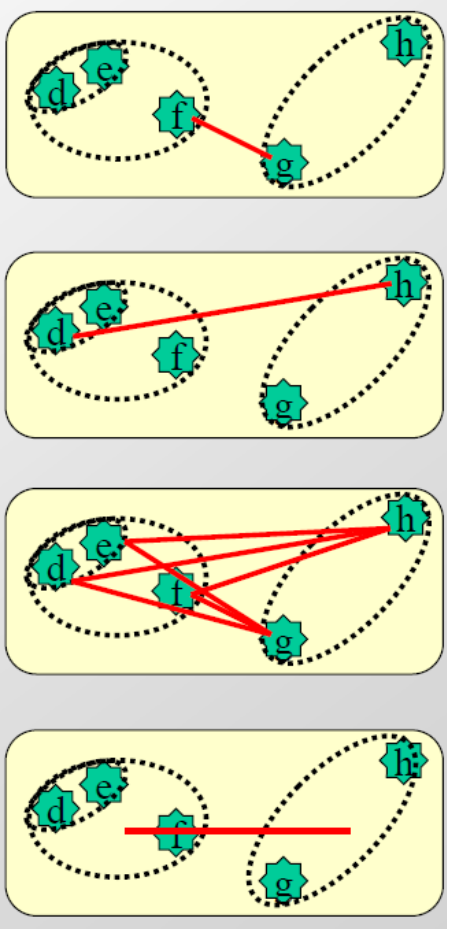
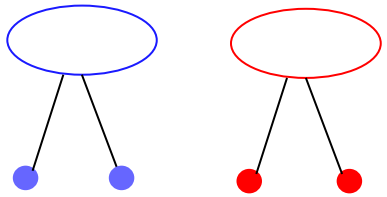


Image from Manolis Kellis

Hierarchical Clustering

1. Initially each point is its own cluster
2. Find pair of clusters with smallest distance between them or equivalently are the most similar
3. Merge into parent cluster
4. Repeat

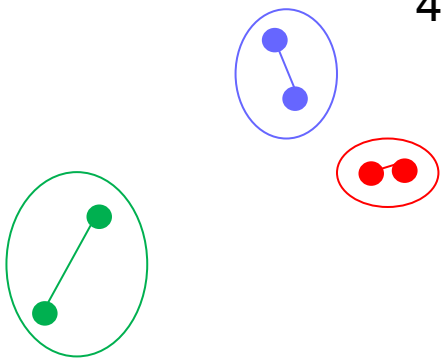
Suppose we continue with complete linkage

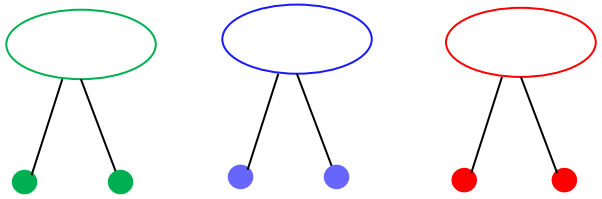


Hierarchical Clustering

1. Initially each point is its own cluster
2. Find pair of clusters with smallest distance between them or equivalently are the most similar
3. Merge into parent cluster
4. Repeat

Suppose we continue with complete linkage

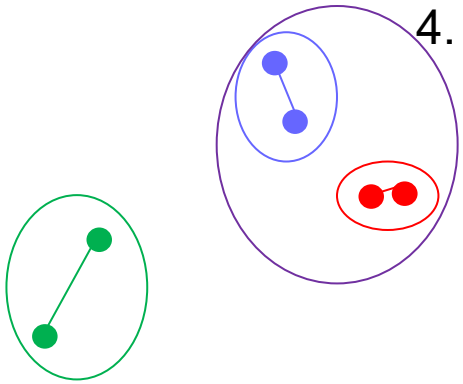


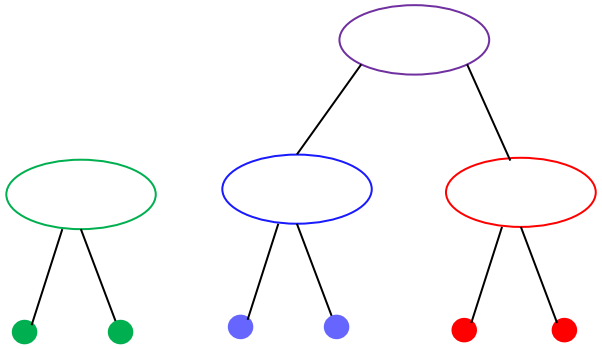


Hierarchical Clustering

1. Initially each point is its own cluster
2. Find pair of clusters with smallest distance between them or equivalently are the most similar
3. Merge into parent cluster
4. Repeat

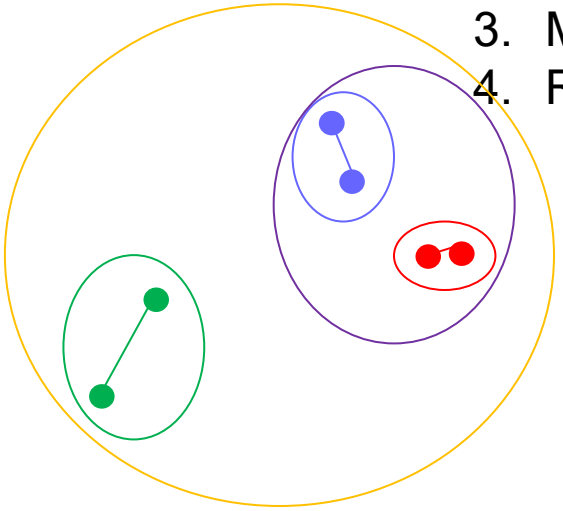
Suppose we continue with complete linkage



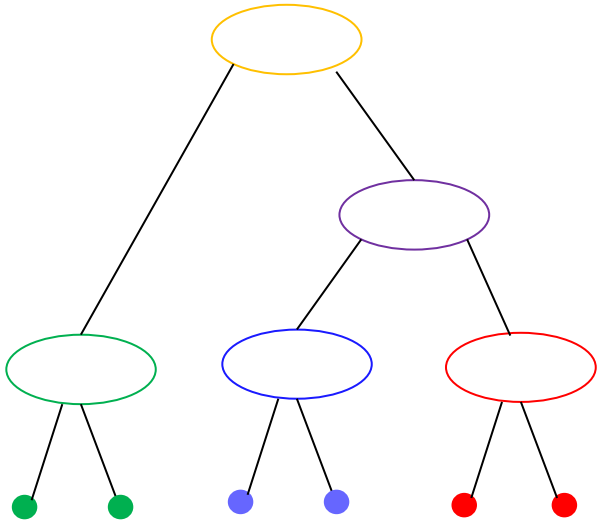


Hierarchical Clustering

1. Initially each point is its own cluster
2. Find pair of clusters with smallest distance between them or equivalently are the most similar
3. Merge into parent cluster
4. Repeat



Suppose we continue with complete linkage



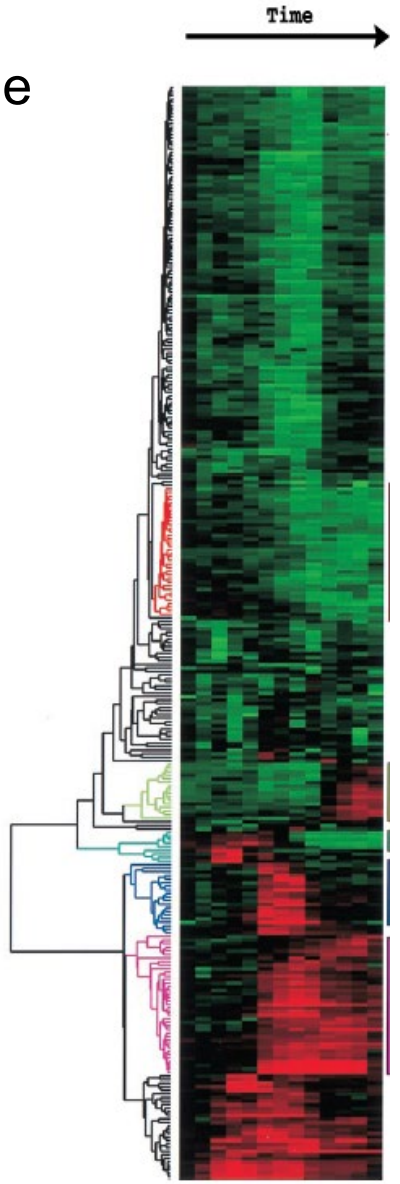
Runtime Complexity of Hierarchical Clustering

- Let n be the number of data points
- $O(n^2)$ time to compute all pairwise distances
- $O(n)$ iterations
- $O(n^3)$ if all pairwise distances recomputed and/or iterated over each iteration
- Depending on details of linkage method and implementation can run in $O(n^2)$ or $O(n^2 \log n)$ time

Ordering Leaves

Often more visual focus goes to the heatmap and the row ordering than the dendogram

Question: Does hierarchical clustering uniquely determine an ordering of leaves (rows)?

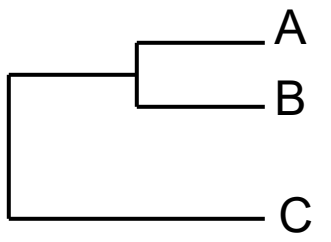


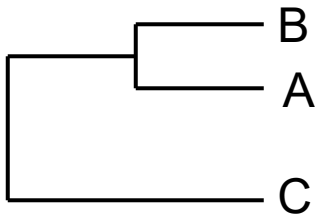
Ordering Leaves

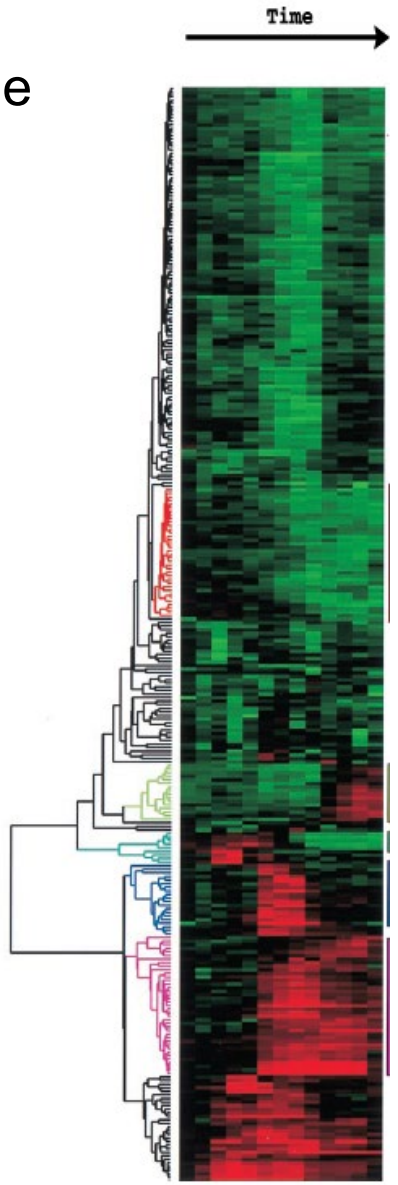
Often more visual focus goes to the heatmap and the row ordering than the dendogram

Question: Does hierarchical clustering uniquely determine an ordering of leaves (rows)?

No!

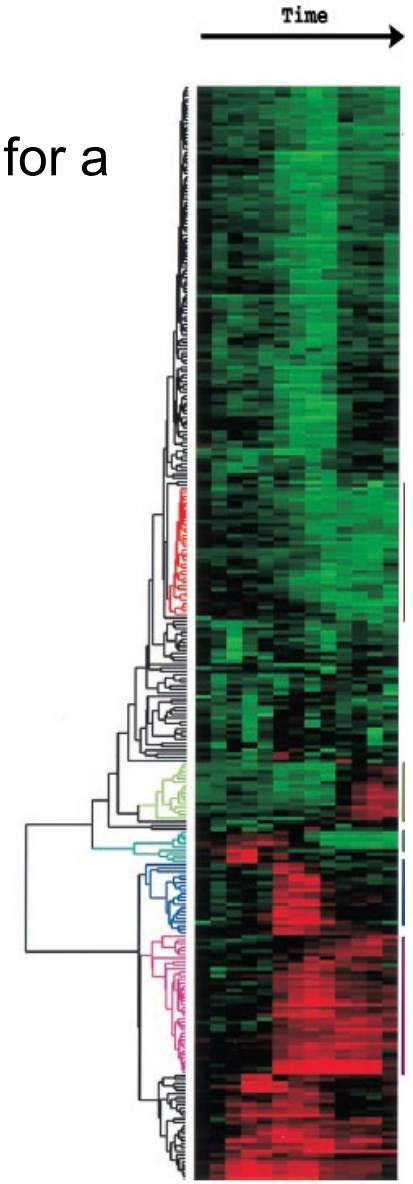






Ordering Leaves

Question: How many possible orderings are there for a hierarchical clustering of *n* data points?

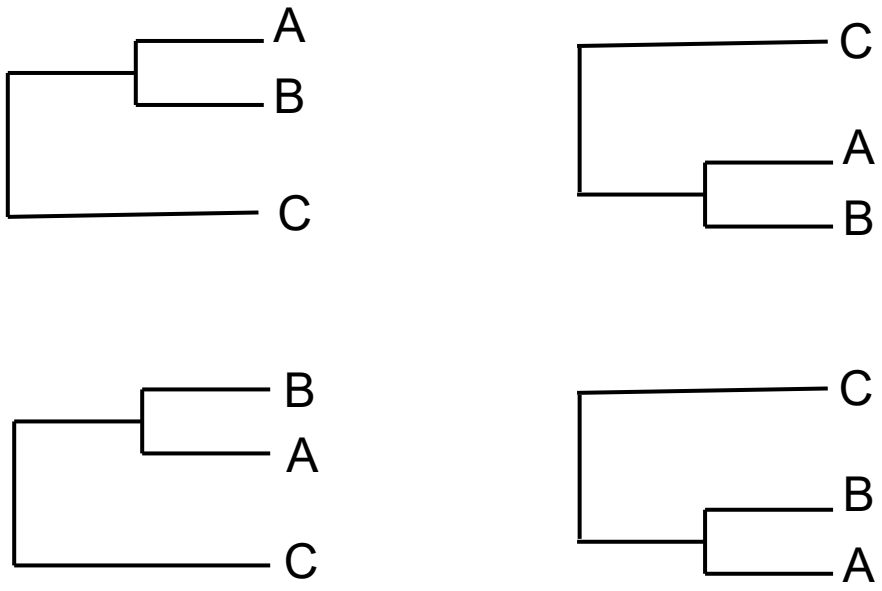


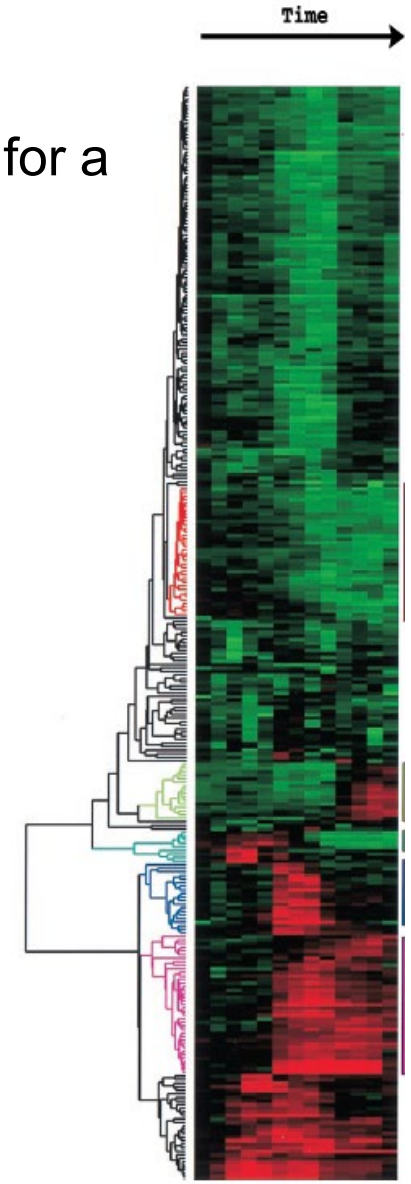
Ordering Leaves

Question: How many possible orderings are there for a hierarchical clustering of *n* data points?

2^{n-1}

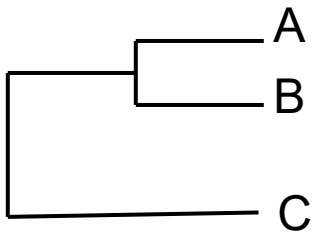
n-1 internal nodes each of which can be flipped

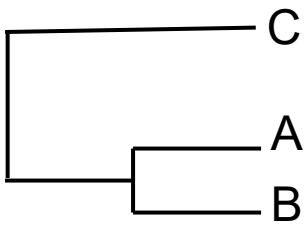


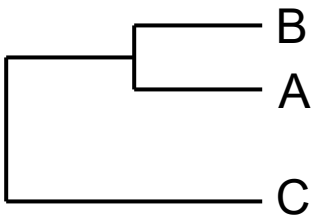


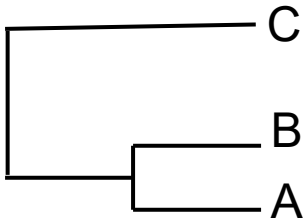
Ordering Leaves

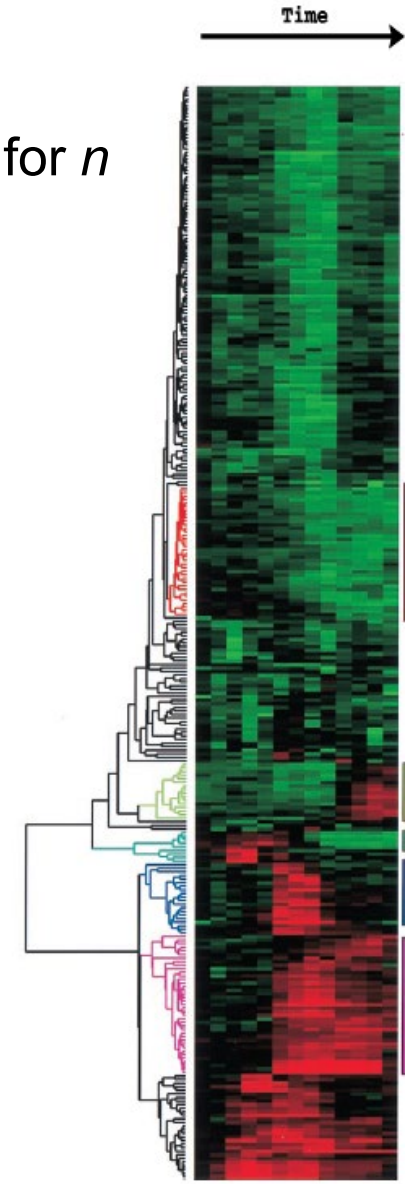
Question: How many possible orderings are there for *n* data points if we were not constrained by the hierarchical clustering?







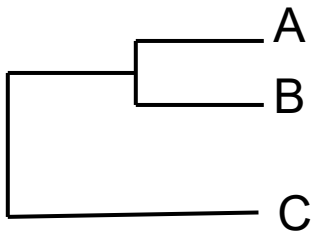


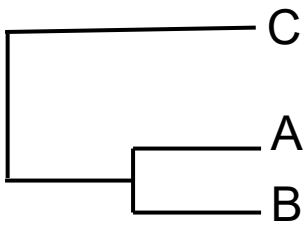


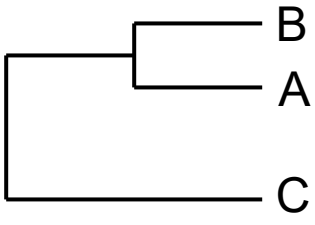
Ordering Leaves

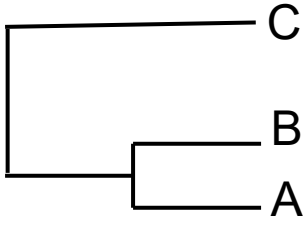
Question: How many possible orderings are there for *n* data points if we were not constrained by the hierarchical clustering?

n!

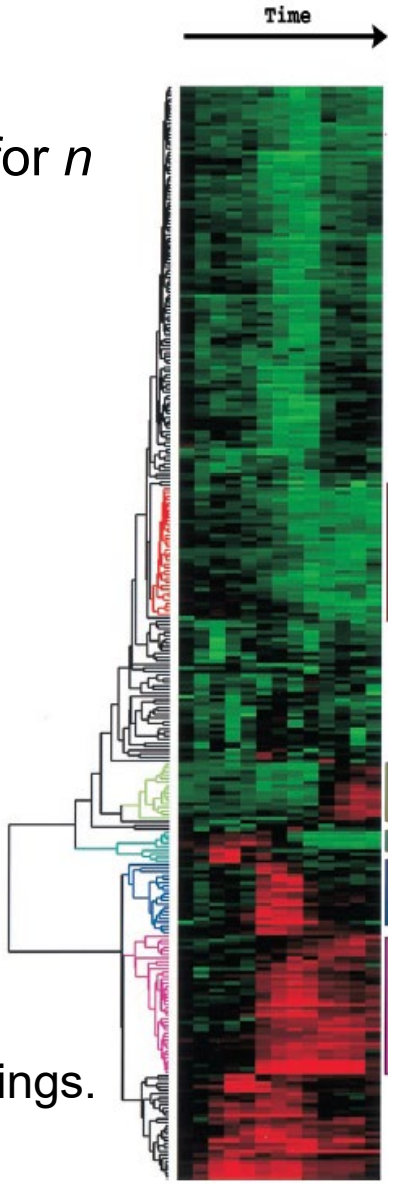








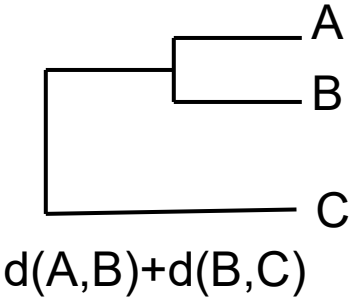
A->C->B and B->C->A would also become possible orderings.



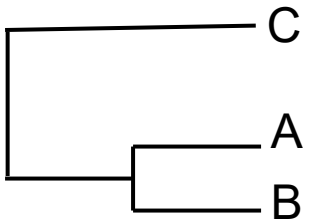
Ordering Leaves

Question: How should we select among possible orderings?

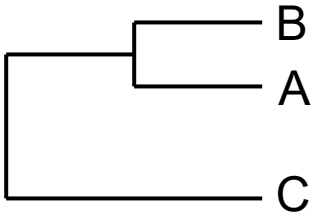
Idea – pick an ordering that minimizes distance between adjacent leaves or equivalently maximizes similarity



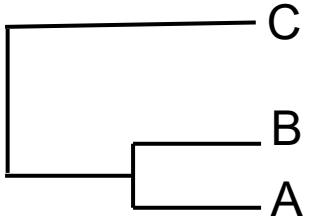
d(A,B)+d(B,C)



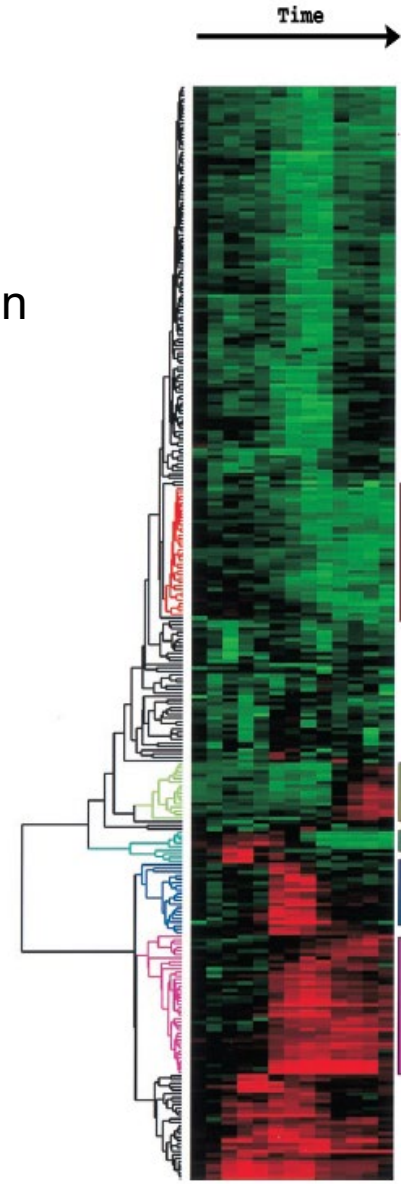
d(C,A)+d(A,B)



d(B,A)+d(A,C)



d(C,B)+d(B,A)



Optimal Ordering Leaves

Problem: Order leaves of hierarchical clustering to minimize sum of distances between neighboring leaves

Question: Brute force trying all possible orderings would take exponential time. Can we efficiently compute an optimal ordering of the leaves and if so how?

Ordering Leaves

Claim in Eisen et al, 1998 paper

Ordering of Data Tables. For any dendrogram of n elements, there are 2^{n-1} linear orderings consistent with the structure of the tree (at each node, either of the two elements joined by the node can be ordered ahead of the other). An optimal linear ordering, one that maximizes the similarity of adjacent elements in the ordering, is impractical to compute.

Proc. Natl. Acad. Sci. USA 95 (1998)

Ordering Leaves

Claim in Eisen et al, 1998 paper

Ordering of Data Tables. For any dendrogram of n elements, there are 2^{n-1} linear orderings consistent with the structure of the tree (at each node, either of the two elements joined by the node can be ordered ahead of the other). An optimal linear ordering, one that maximizes the similarity of adjacent elements in the ordering, is impractical to compute.

Proc. Natl. Acad. Sci. USA 95 (1998)

Burkard et al, Mathematics of Operations Research, 1998 O(n^3)
Bar-Joseph et al, *Bioinformatics* 2001 O(n^4)
Bar-Joseph et al, *Bioinformatics* 2003 O(n^3)

Optimal Ordering Leaves in $O(n^3)$ time

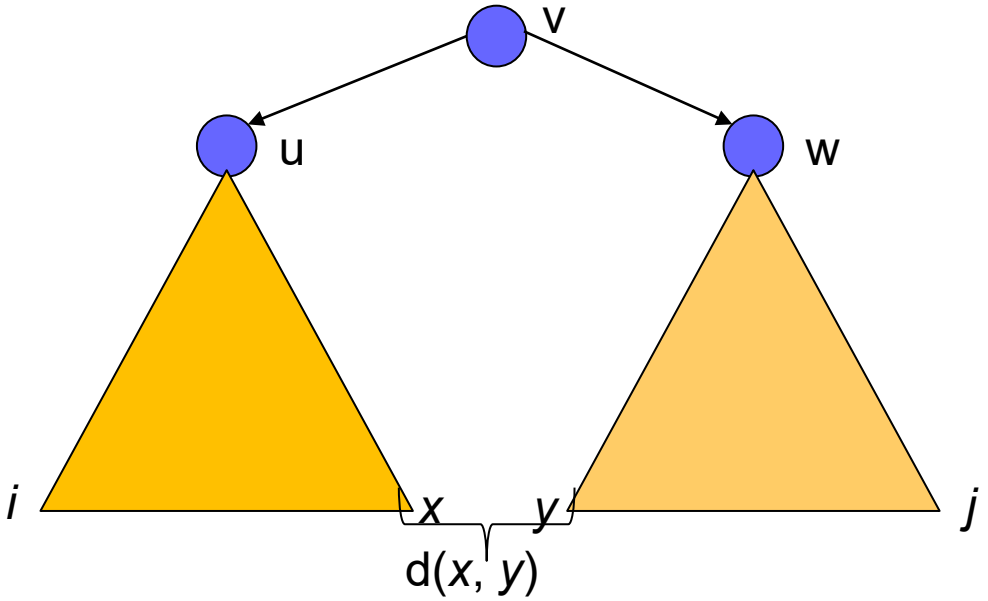
T(*v*) – subtree rooted at *v*

M(*v*, *i*, *j*) – cost of optimal tree rooted at *v* with start and end leaves *i* and *j* respectively where *i*,*j* are leaves in T(*v*)

$M(v, v, v) = 0$ *v* – is a leaf

$$M(v, i, j) = \min_{x \in T(u), y \in T(w)} M(u, i, x) + d(x, y) + M(w, y, j)$$

u – is left sub-child
w – is right sub-child



Optimal Ordering Leaves in $O(n^3)$ time

T(*v*) – subtree rooted at *v*

M(*v*, *i*, *j*) – cost of optimal tree rooted at *v* with start and end leaves *i* and *j* respectively where *i*,*j* are leaves in T(*v*)

$M(v, v, v) = 0$ *v* – is a leaf

$$M(v, i, j) = \min_{x \in T(u), y \in T(w)} M(u, i, x) + d(x, y) + M(w, y, j)$$

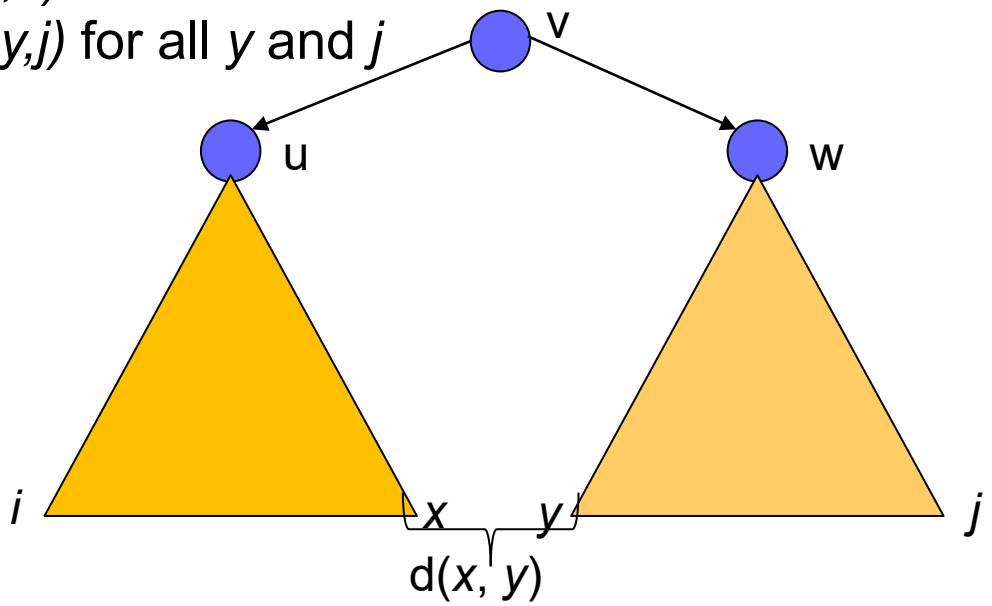
u – is left sub-child
w – is right sub-child

If v is a leaf M(v,v,v) = 0 otherwise
Recursively compute and store M(*u*,*i*,*x*) for all *i* and *x*
Recursively compute and store M(*w*,*y*,*j*) for all *y* and *j*
for *i* leaves in T(*u*)
  for *y* leaves in T(*w*)

$$temp_i[y] = \min_{x \in T(u)} M(u,i,x)+d(x,y)$$

  for *j* leaves in T(*w*)

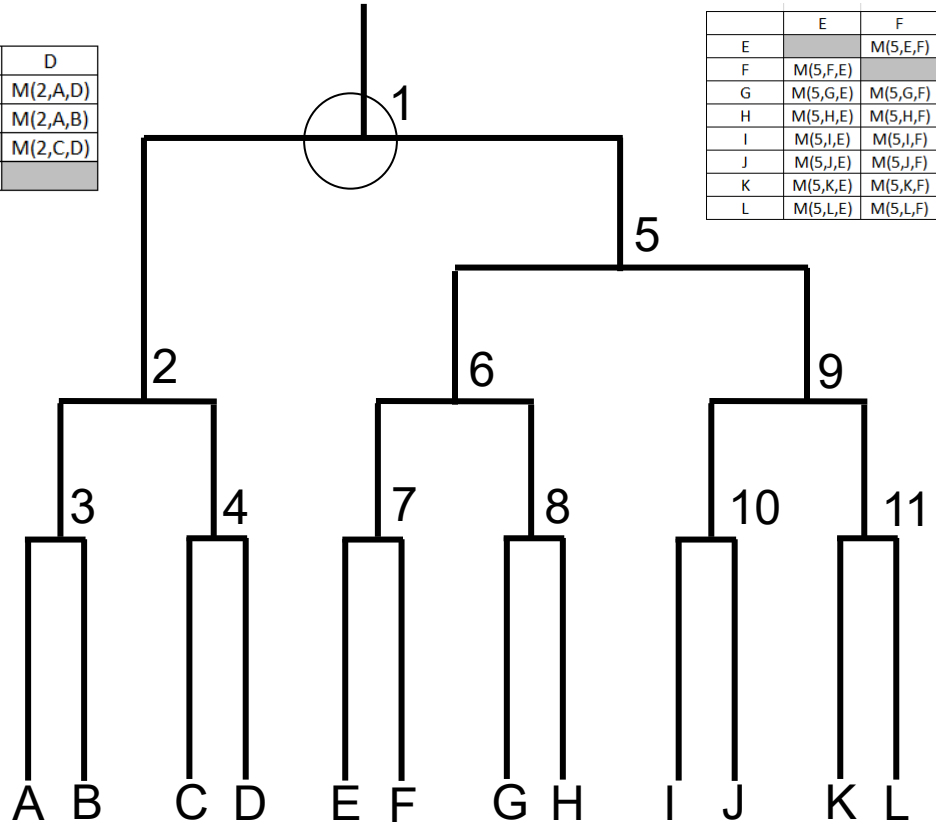
$$M(v, i, j) = \min_{y \in T(w)} temp_i[y]+M(w,y,j)$$



Optimal Ordering Leaves in $O(n^3)$ time

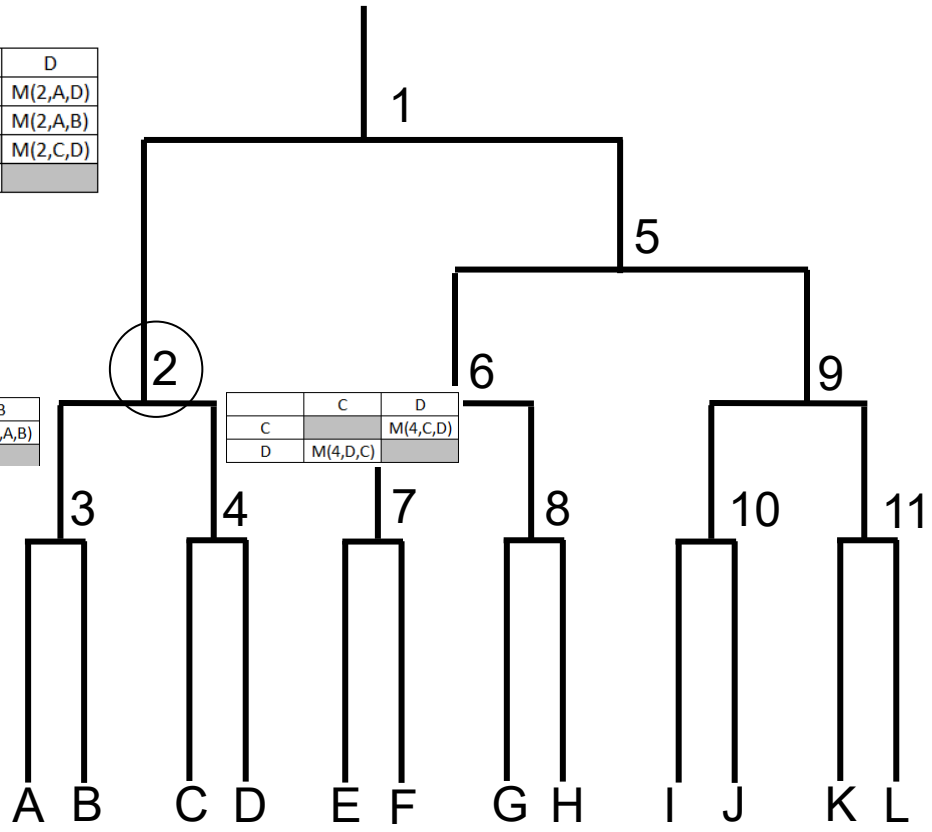
	A	B	C	D
A		M(2,A,B)	M(2,A,C)	M(2,A,D)
B	M(2,B,A)		M(2,A,B)	M(2,A,B)
C	M(2,C,A)	M(2,C,B)		M(2,C,D)
D	M(2,D,A)	M(2,D,B)	M(2,D,C)	

	E	F	G	H	I	J	K	L
E		M(5,E,F)	M(5,E,G)	M(5,E,H)	M(5,E,I)	M(5,E,J)	M(5,E,K)	M(5,E,L)
F	M(5,F,E)		M(5,F,G)	M(5,F,H)	M(5,F,I)	M(5,F,J)	M(5,F,K)	M(5,F,L)
G	M(5,G,E)	M(5,G,F)		M(5,G,H)	M(5,G,I)	M(5,G,J)	M(5,G,K)	M(5,G,L)
H	M(5,H,E)	M(5,H,F)	M(5,H,G)		M(5,H,I)	M(5,H,J)	M(5,H,K)	M(5,H,L)
I	M(5,I,E)	M(5,I,F)	M(5,I,G)	M(5,I,H)		M(5,I,J)	M(5,I,K)	M(5,I,L)
J	M(5,J,E)	M(5,J,F)	M(5,J,G)	M(5,J,H)	M(5,J,I)		M(5,J,K)	M(5,J,L)
K	M(5,K,E)	M(5,K,F)	M(5,K,G)	M(5,K,H)	M(5,K,I)	M(5,K,J)		M(5,K,L)
L	M(5,L,E)	M(5,L,F)	M(5,L,G)	M(5,L,H)	M(5,L,I)	M(5,L,J)	M(5,L,K)	



Optimal Ordering Leaves in $O(n^3)$ time

	A	B	C	D
A		M(2,A,B)	M(2,A,C)	M(2,A,D)
B	M(2,B,A)		M(2,A,B)	M(2,A,B)
C	M(2,C,A)	M(2,C,B)		M(2,C,D)
D	M(2,D,A)	M(2,D,B)	M(2,D,C)	



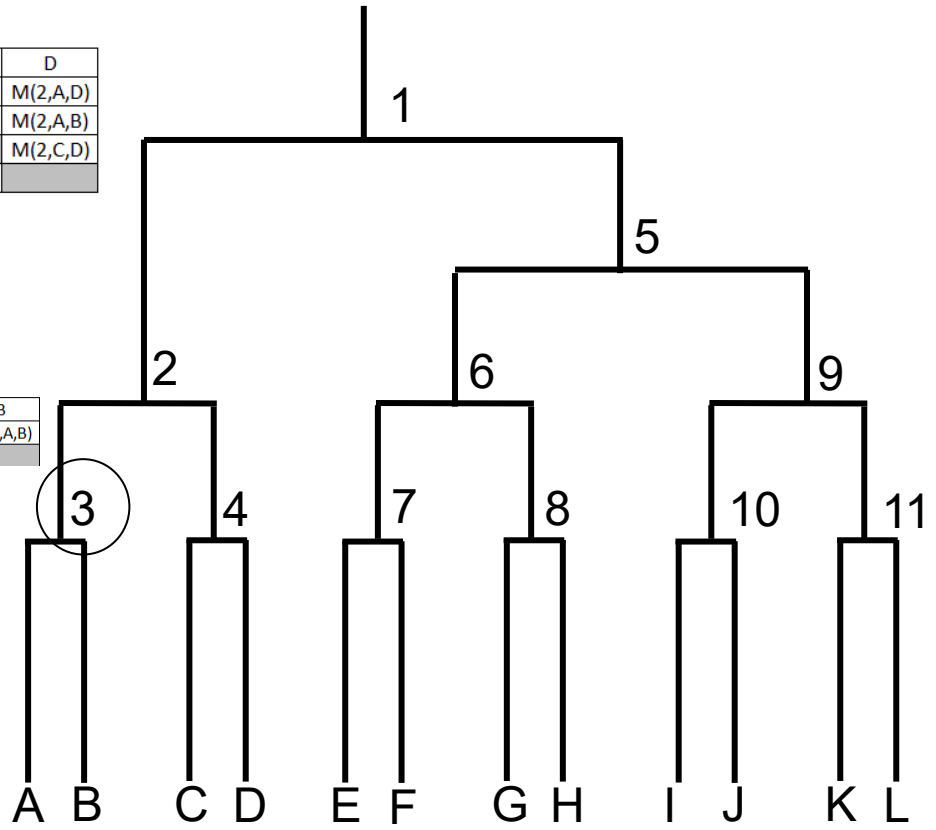
	A	B
A		M(3,A,B)
B	M(3,B,A)	

	C	D
C		M(4,C,D)
D	M(4,D,C)	

Optimal Ordering Leaves in $O(n^3)$ time

	A	B	C	D
A		M(2,A,B)	M(2,A,C)	M(2,A,D)
B	M(2,B,A)		M(2,A,B)	M(2,A,B)
C	M(2,C,A)	M(2,C,B)		M(2,C,D)
D	M(2,D,A)	M(2,D,B)	M(2,D,C)	

	A	B
A		M(3,A,B)
B	M(3,B,A)	



M(A,A,A)=0

M(B,B,B)=0

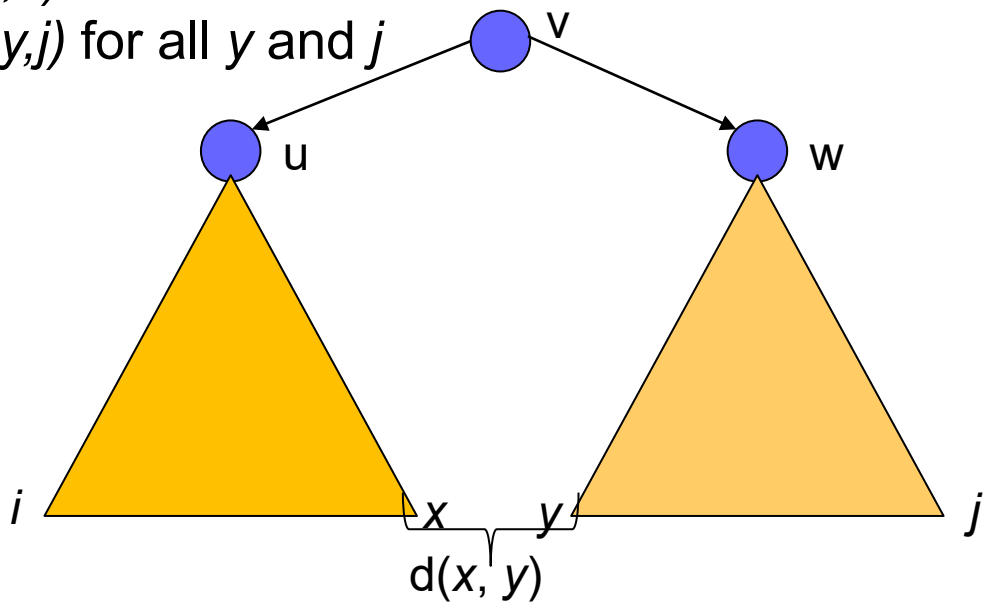
Optimal Ordering Leaves in $O(n^3)$ time

Let F(n) be the total time to compute M(v, i, j) for all i, j in a tree with n leaves
Let r be number of leaves in subtree T(u)
Let s be number of leaves in subtree T(w)

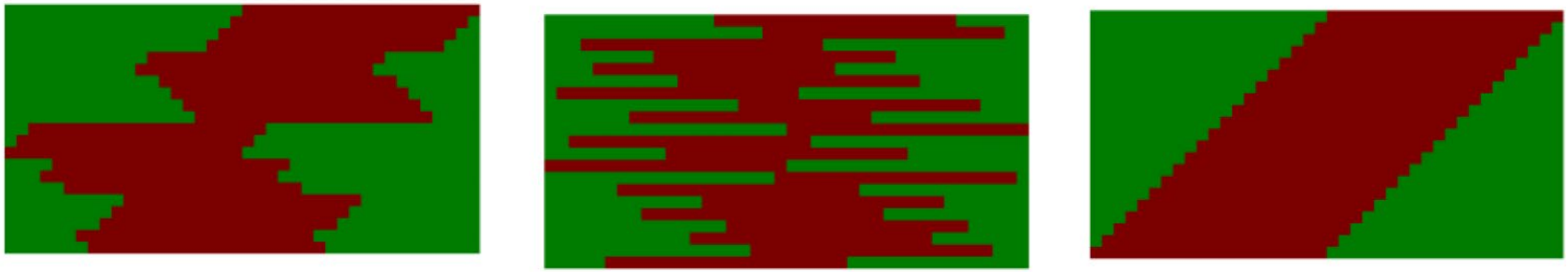
$F(n) = F(r) + F(s) + O(r^2 s) + O(r s^2)$
We have $r+s = n$ and
$(r^3 + s^3 + r^2 s + r s^2) <= (r+s)^3 = n^3$
By induction it follows that F(n) is $O(n^3)$

If v is a leaf M(v,v,v) = 0 otherwise
Recursively compute and store M(u,i,x) for all i and x
Recursively compute and store M(w,y,j) for all y and j
for i leaves in T(u)
  for y leaves in T(w)
    $temp_i[y] = \min_{x \in T(u)} M(u,i,x)+d(x,y)$

  for j leaves in T(w)
    $M(v, i, j) = \min_{y \in T(w)} temp_i[y]+M(w,y,j)$



Ordering Rows Optimally



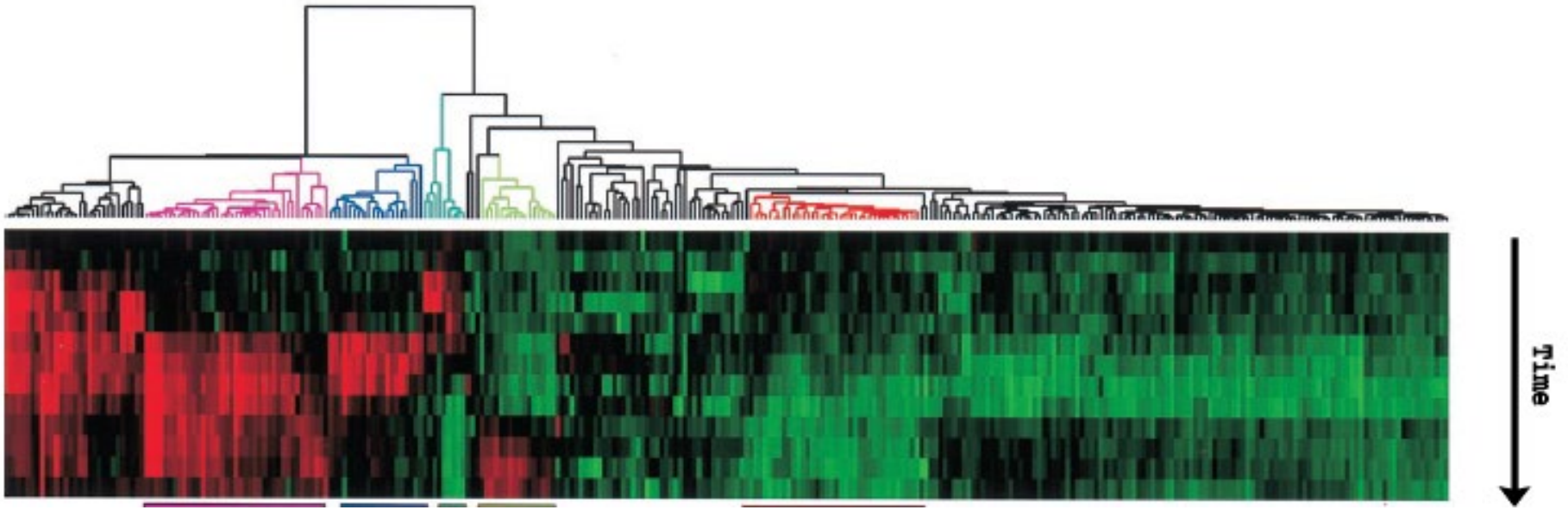
Cluster result | Input (permuted) | Ordering result

Rows are genes. Columns experiments
Green corresponds to -1 values
Red corresponds to 1 values

Image from Bar-Joseph et al *Bioinformatics* 2001

Getting Clusters from Hierarchical Clustering

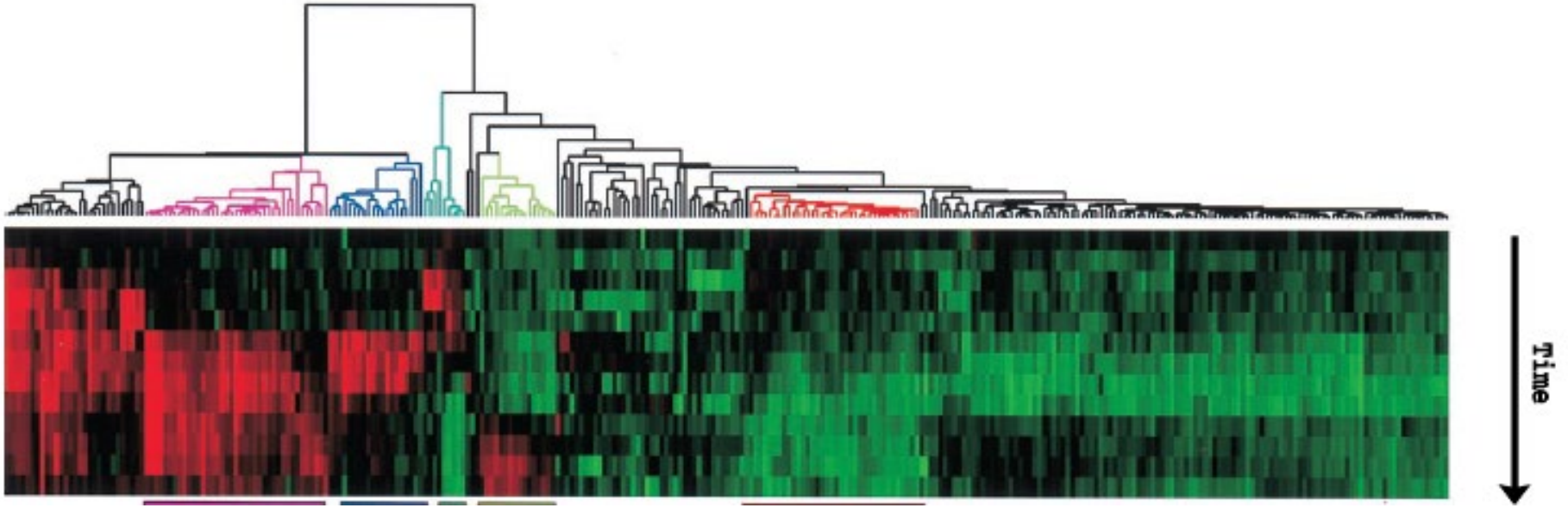
How can we get *k* clusters from hierarchical clustering?



Getting Clusters from Hierarchical Clustering

How can we get *k* clusters from hierarchical clustering?

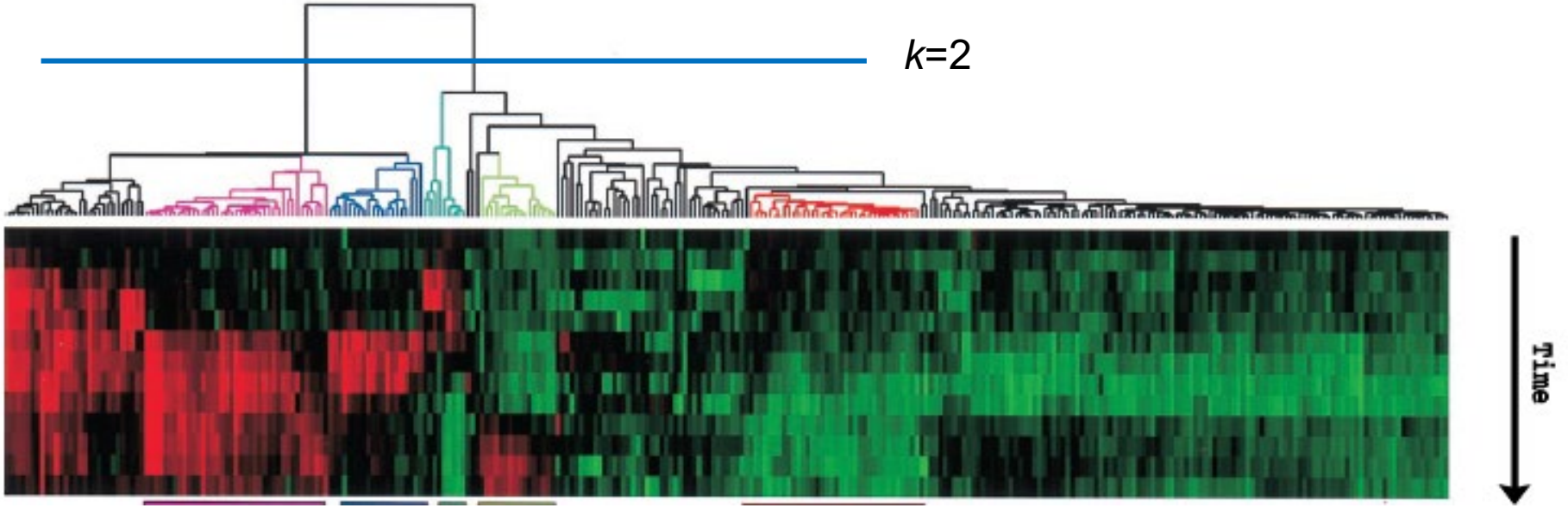
Idea – cut tree to undo last k-1 merges



Getting Clusters from Hierarchical Clustering

How can we get *k* clusters from hierarchical clustering?

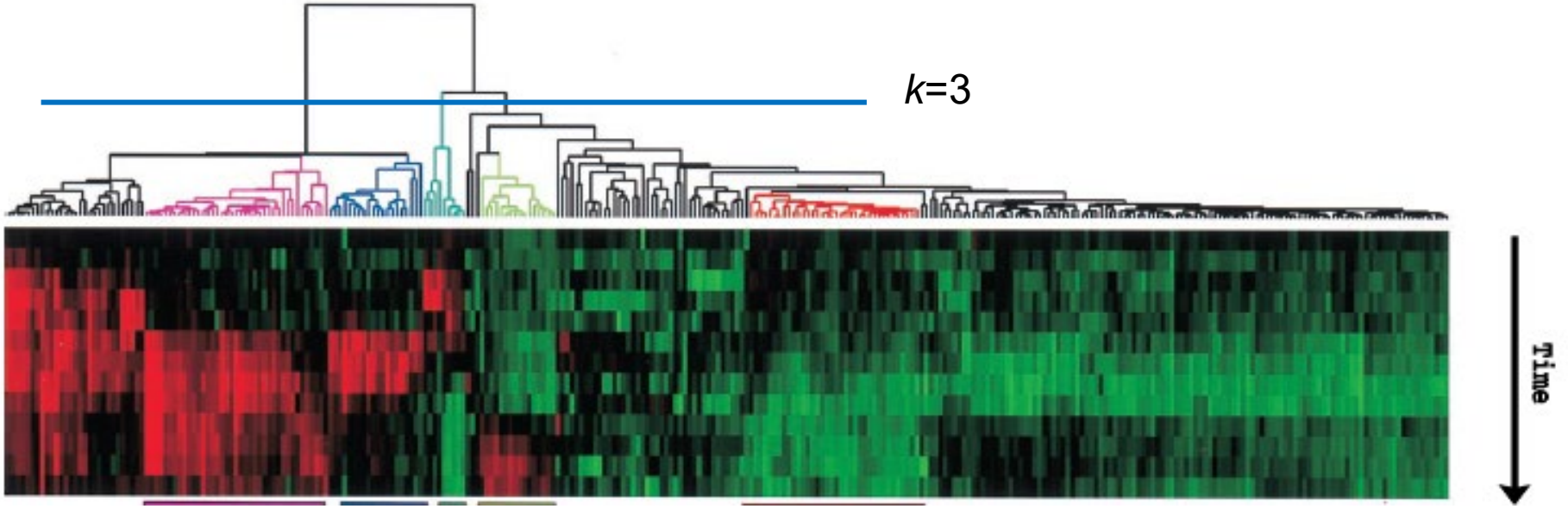
Idea – cut tree to undo last k-1 merges



Getting Clusters from Hierarchical Clustering

How can we get *k* clusters from hierarchical clustering?

Idea – cut tree to undo last k-1 merges



- Popular since show all the data in a hierarchy, but limited theoretical basis for resulting clusters (i.e. no associated optimization criteria or statistical model)

k-means clustering

Lecture 4

April 13th, 2023

Based on slides from Andrew Moore

K-means Objective Function

$$\arg\min \sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - \mu_i||^2$$

μ_i Mean of cluster *i*

S_i Data points assigned to cluster *i*

K-means Objective Function

$$\arg\min \sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - \mu_i||^2$$

μ_i Mean of cluster *i*

S_i Data points assigned to cluster *i*

Can be motivated by minimizing loss of information in compression

- an encoder function: ENCODE : $\Re^d \rightarrow [1..k]$
- a decoder function: DECODE : $[1..k] \rightarrow \Re^d$

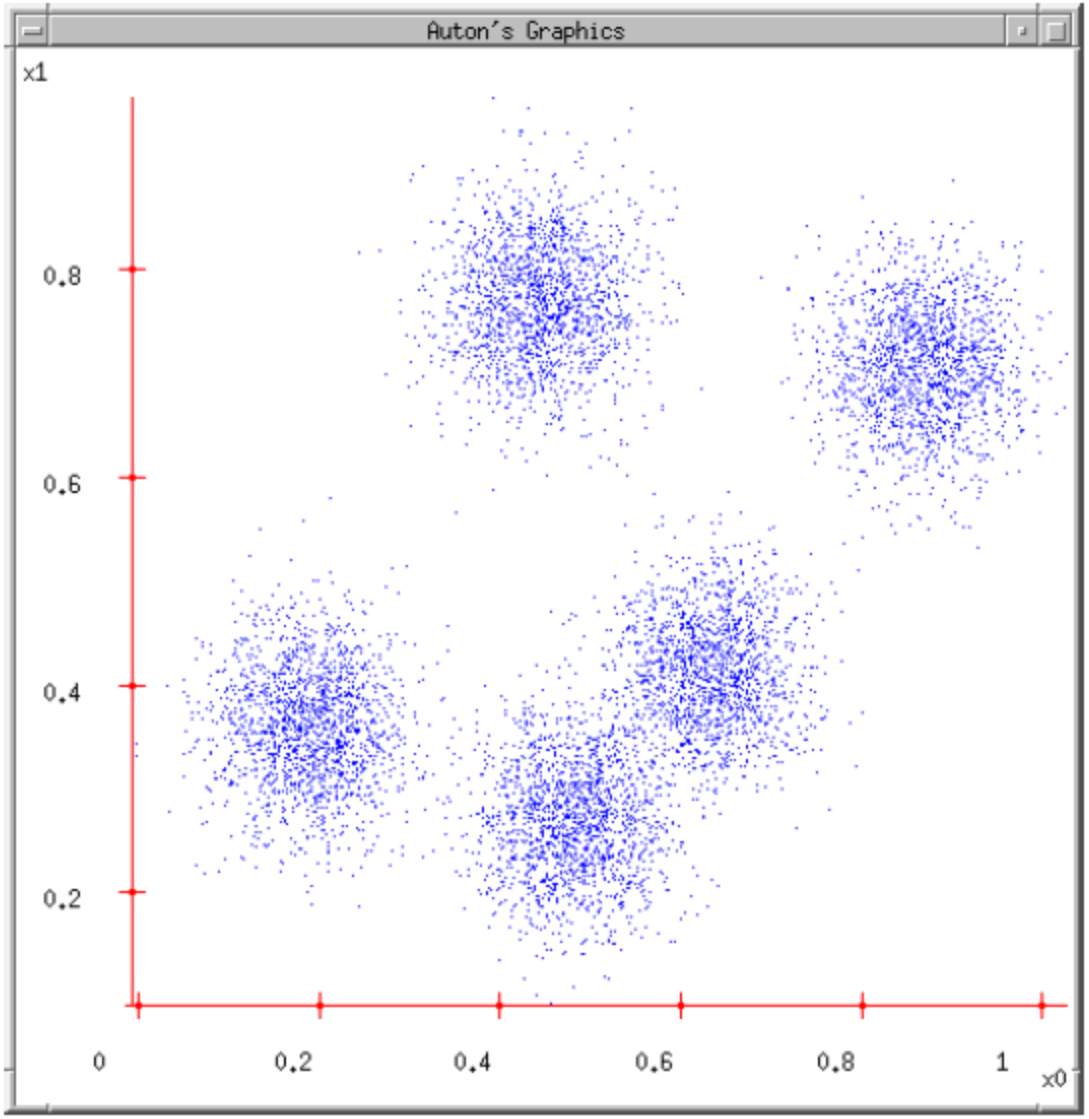
Define...

$$\text{Distortion} = \sum_{i=1}^{n} \left(\mathbf{x}_i - \text{DECODE}[\text{ENCODE}(\mathbf{x}_i)]\right)^2$$

K-means algorithm example

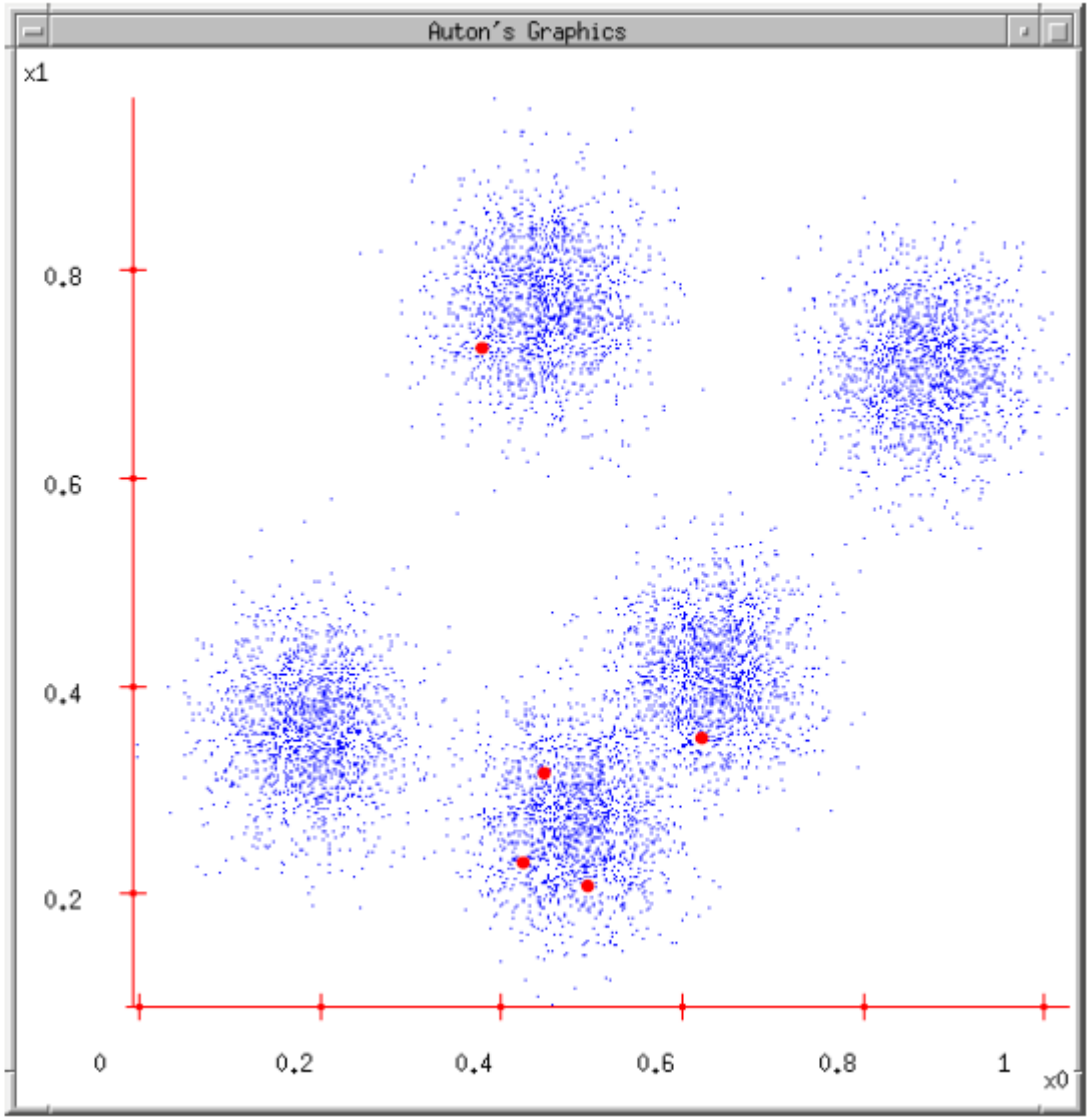
Same as Lloyd's algorithm

1. Ask user how many clusters they'd like. *(e.g. k=5)*



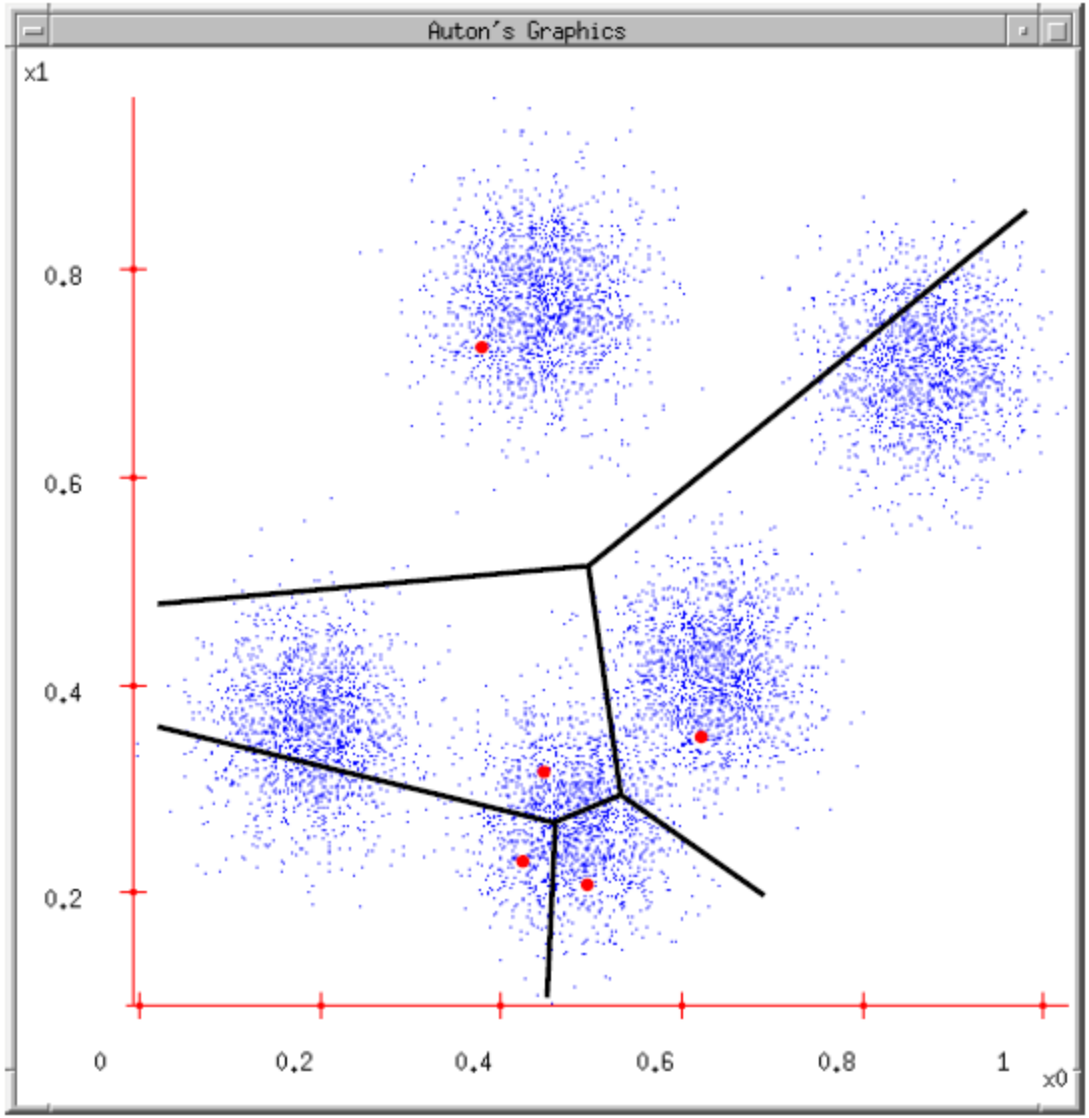
K-means algorithm example

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



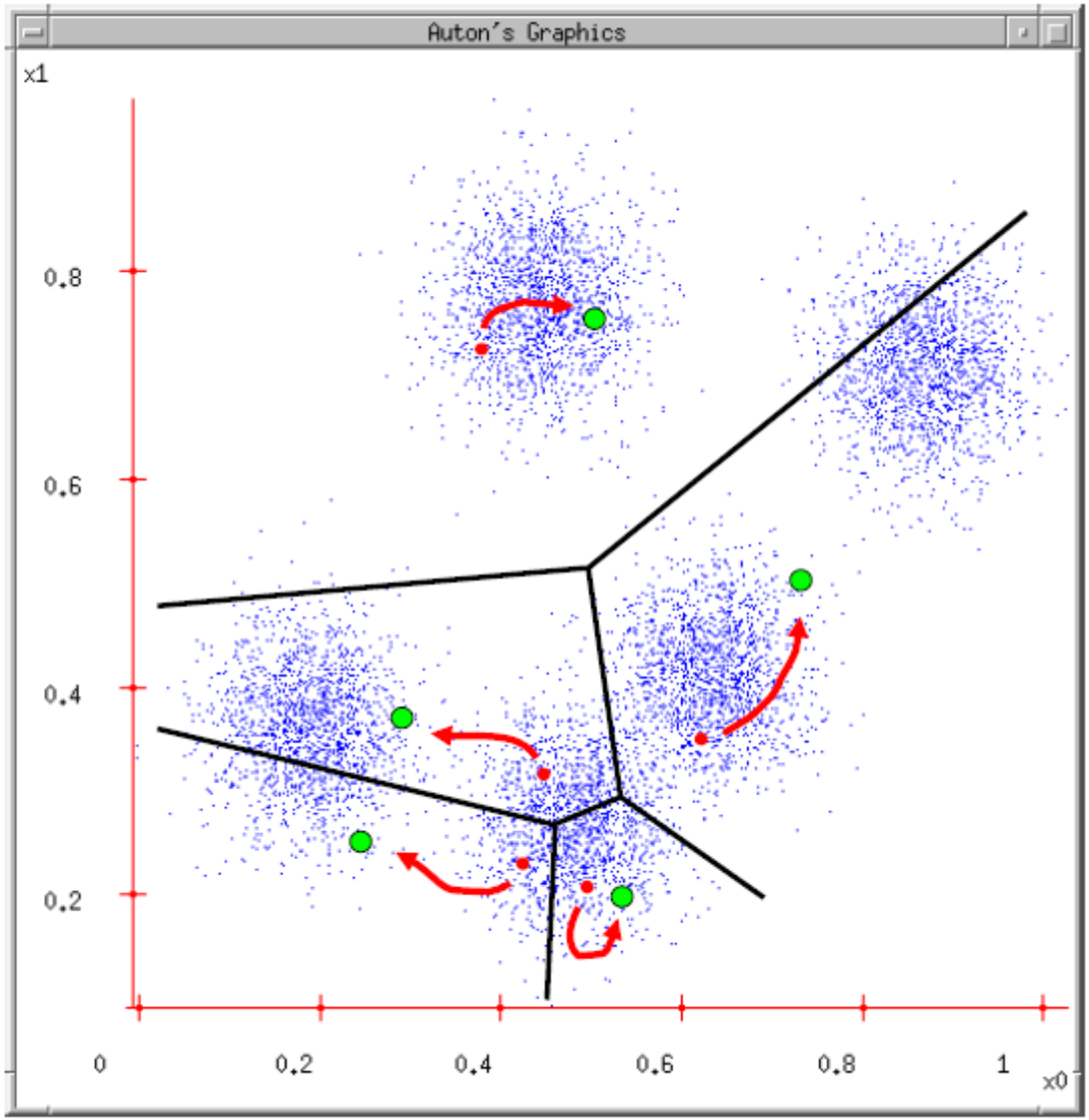
K-means algorithm example

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



K-means algorithm example

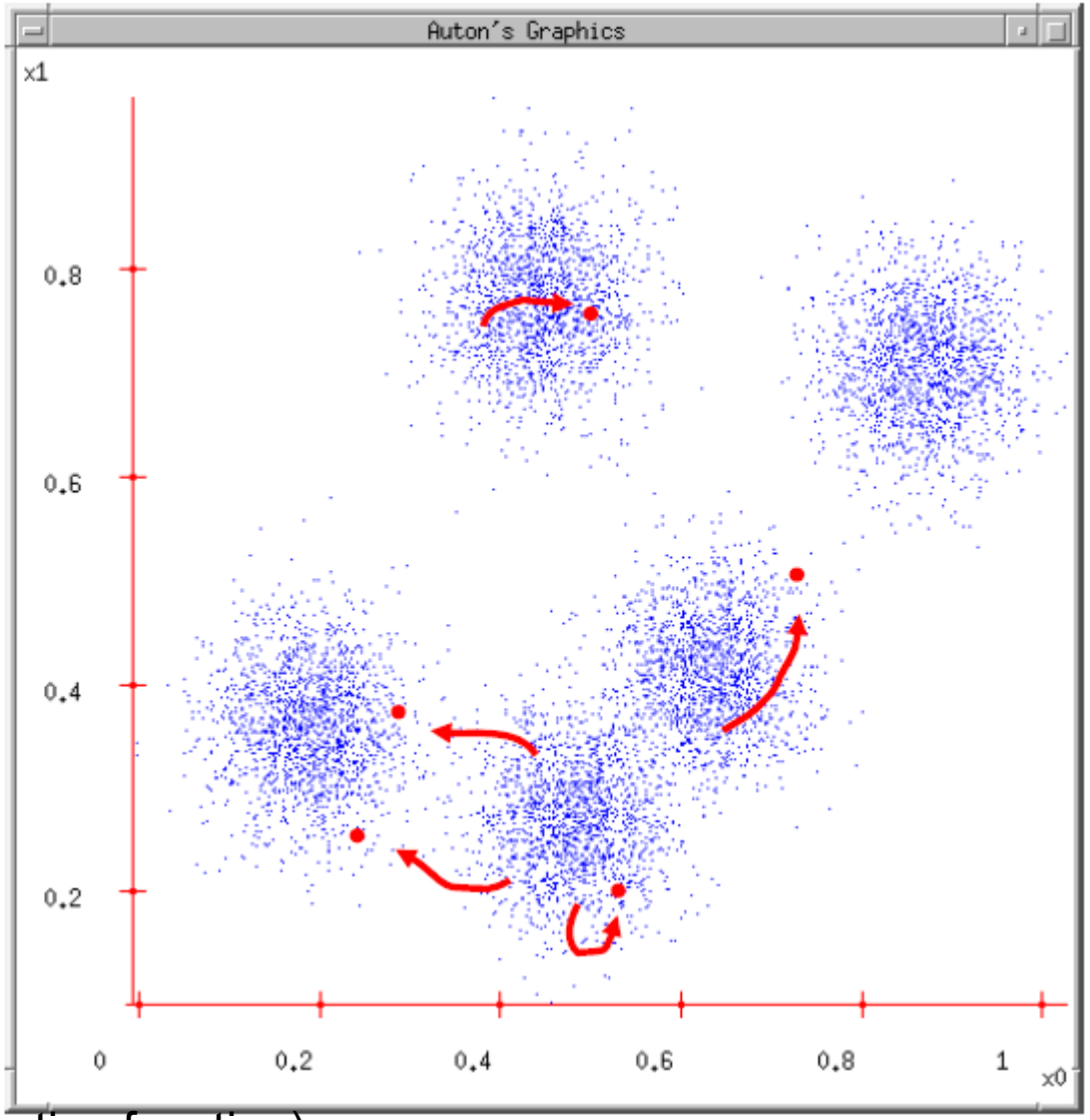
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



K-means algorithm example

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!

(no improvement in objective function)



K-means Convergence

Is the k-means algorithm guaranteed to converge (i.e. no change in objective cost)?

K-means Convergence

Is the k-means algorithm guaranteed to converge (i.e. no change in objective cost)?

Yes. The algorithm will converge since there are only a finite number of ways of partitioning the set of data points into k-groups. Each iteration would need to visit a new configuration since cannot decrease objective value between iterations.

K-means Convergence

Is the k-means algorithm guaranteed to converge (i.e. no change in objective cost)?

Yes. The algorithm will converge since there are only a finite number of ways of partitioning the set of data points into k-groups. Each iteration would need to visit a new configuration since cannot decrease objective value between iterations.

Is the k-means algorithm guaranteed to find an optimal solution?

K-means Convergence

Is the k-means algorithm guaranteed to converge (i.e. no change in objective cost)?

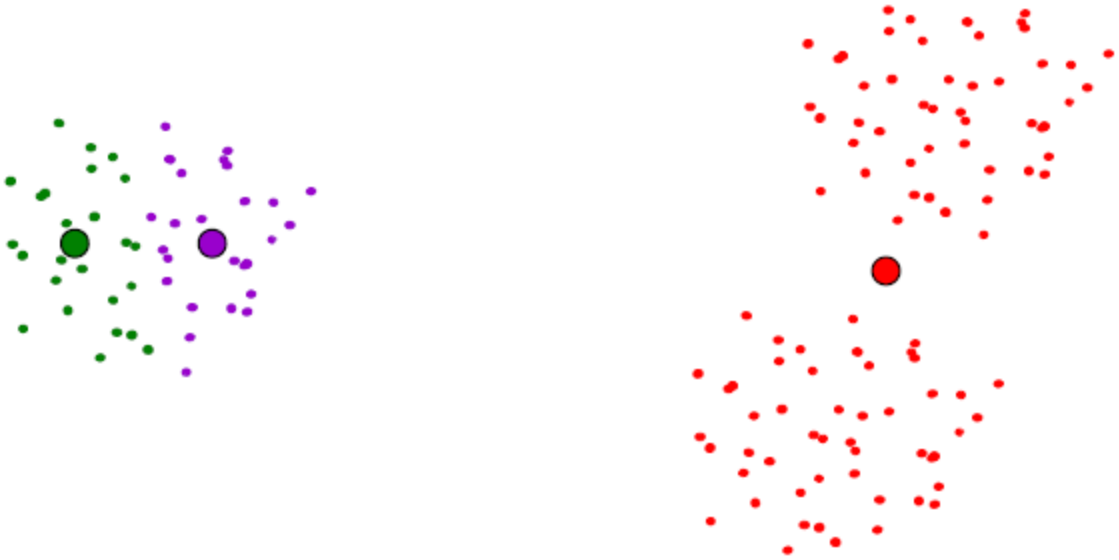
Yes. The algorithm will converge since there are only a finite number of ways of partitioning the set of data points into k-groups. Each iteration would need to visit a new configuration since cannot decrease objective value between iterations.

Is the k-means algorithm guaranteed to find an optimal solution?

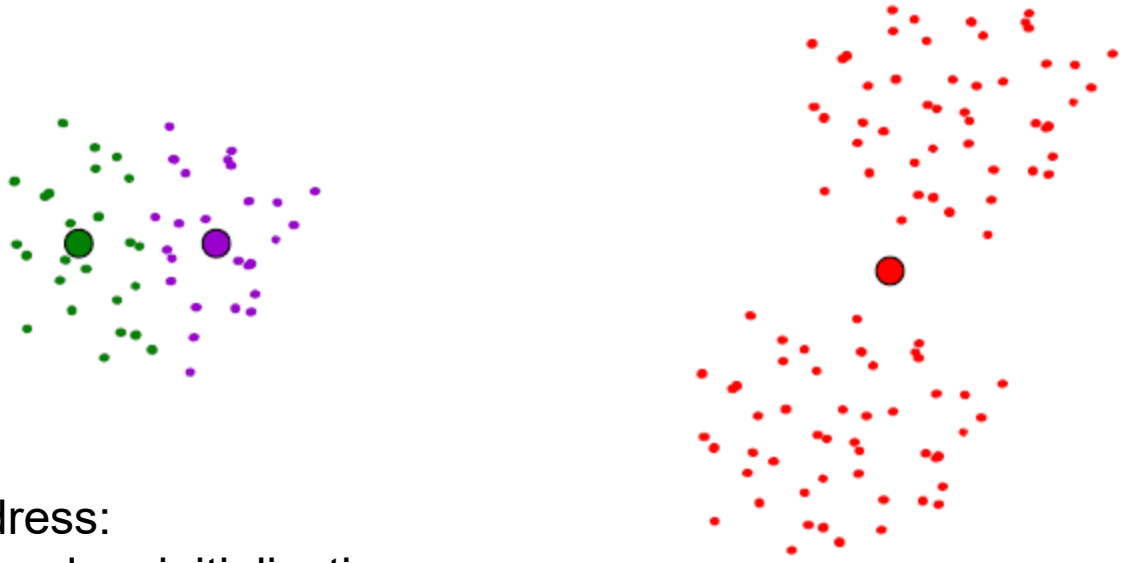
No.

Initialization leading to non-optimal configuration

Initialization leading to non-optimal configuration



Initialization leading to non-optimal configuration



Ways to address:

- Multiple random initializations
- Initialize the clustering in a smarter way

How should we initialize the clustering in a smarter way?

One idea – Furthest Point Heuristic

Furthest point heuristic:

- Pick first cluster center to be one arbitrarily selected point
- Iteratively pick cluster centers to be remaining points that are furthest from any selected point

One idea – Furthest Point Heuristic

Furthest point heuristic:

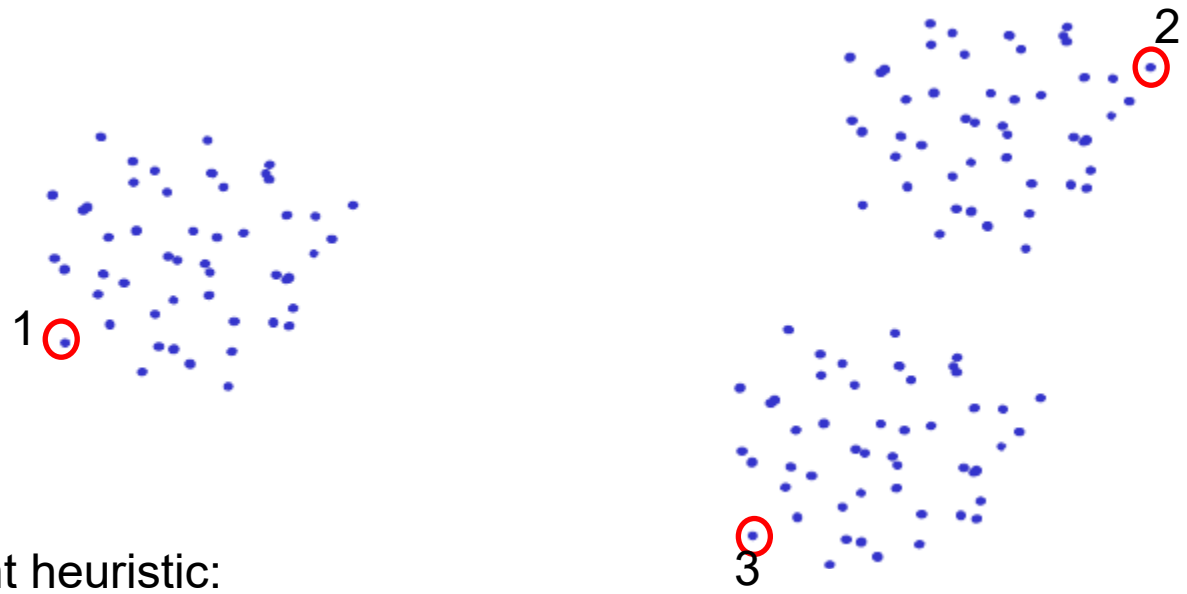
- Pick first cluster center to be one arbitrarily selected point
- Iteratively pick cluster centers to be remaining points that are furthest from any selected point

One idea – Furthest Point Heuristic

Furthest point heuristic:

- Pick first cluster center to be one arbitrarily selected point
- Iteratively pick cluster centers to be remaining points that are furthest from any selected point

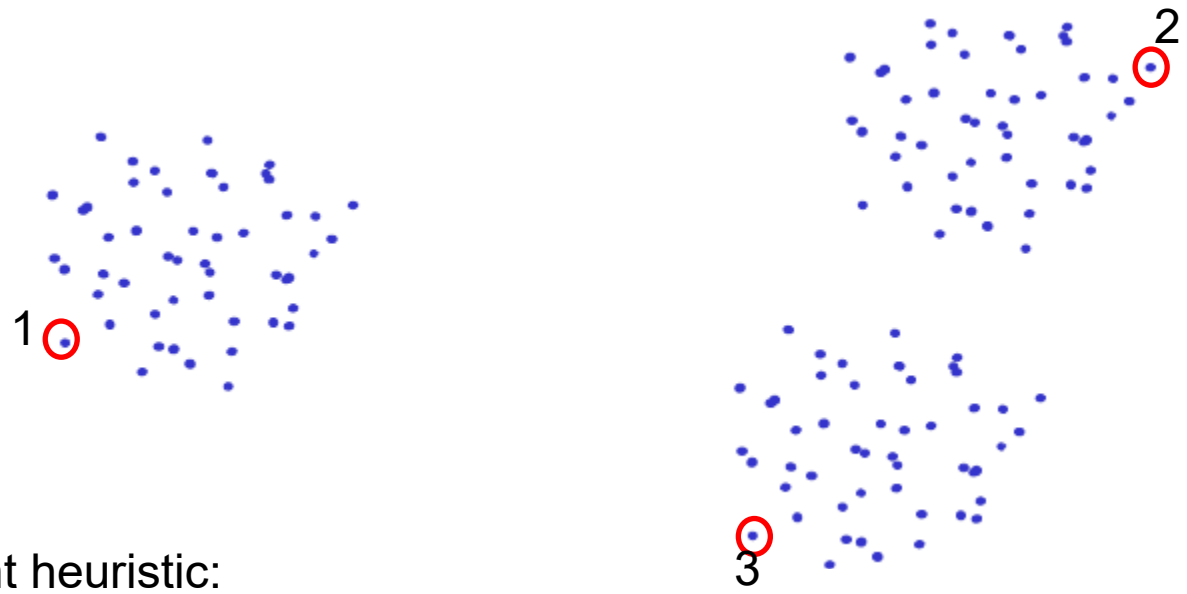
One idea – Furthest Point Heuristic



Furthest point heuristic:

- Pick first cluster center to be one arbitrarily selected point
- Iteratively pick cluster centers to be remaining points that are furthest from any selected point

One idea – Furthest Point Heuristic



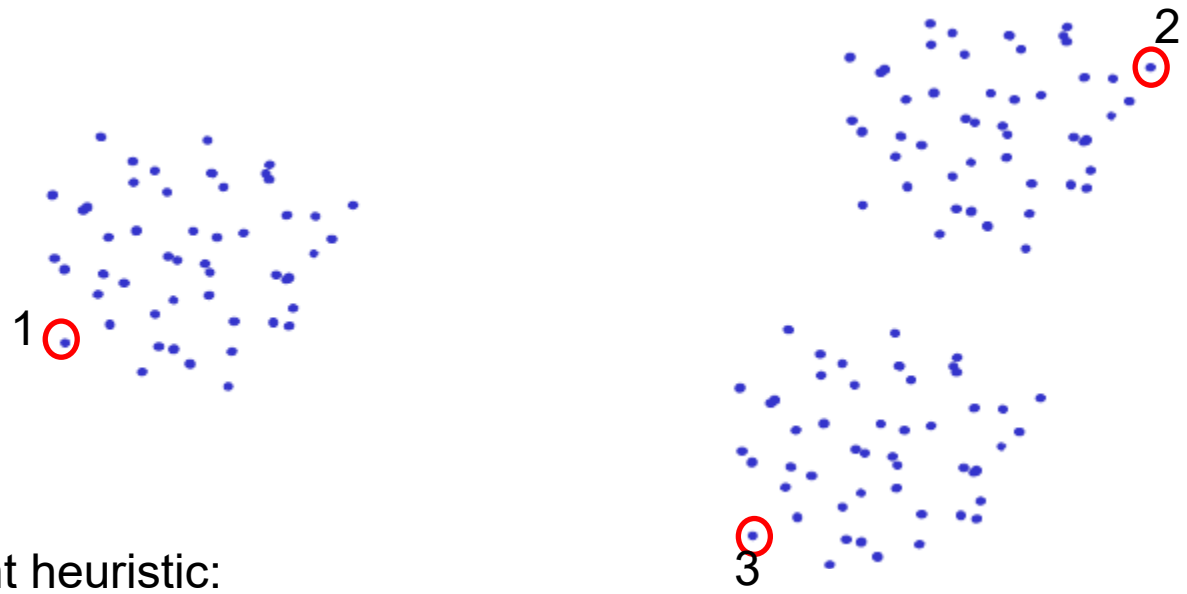
Furthest point heuristic:

- Pick first cluster center to be one arbitrarily selected point
- Iteratively pick cluster centers to be remaining points that are furthest from any selected point

Approximation algorithm to k-centers problem – minimize maximum distance of any point to its closest center

One idea – Furthest Point Heuristic

Question: Can this fail to lead to a good clustering?



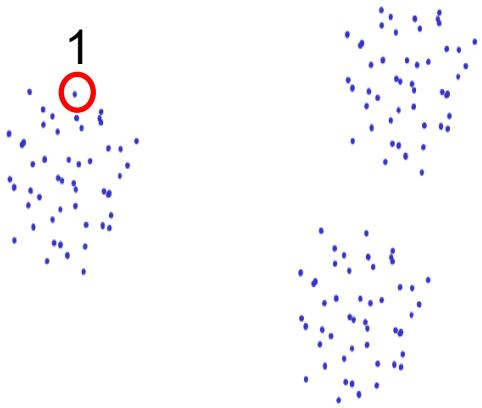
Furthest point heuristic:

- Pick first cluster center to be one arbitrarily selected point
- Iteratively pick cluster centers to be remaining points that are furthest from any selected point

Approximation algorithm to k-centers problem – minimize maximum distance of any point to its closest center

One idea – Furthest Point Heuristic

Question: Can this fail to lead to a good clustering?



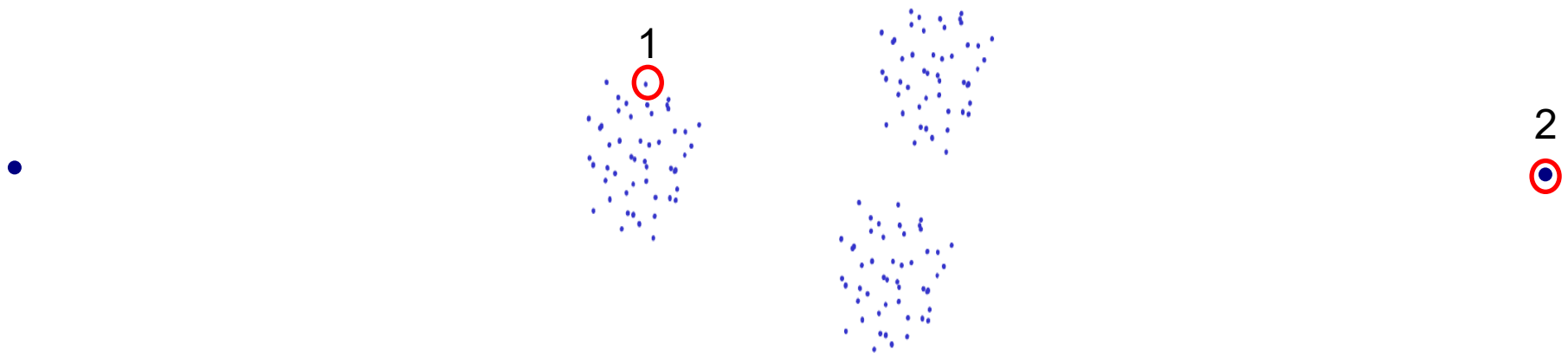
Furthest point heuristic:

- Pick first cluster center to be one arbitrarily selected point
- Iteratively pick cluster centers to be remaining points that are furthest from any selected point

Approximation algorithm to k-centers problem – minimize maximum distance of any point to its closest center

One idea – Furthest Point Heuristic

Question: Can this fail to lead to a good clustering?



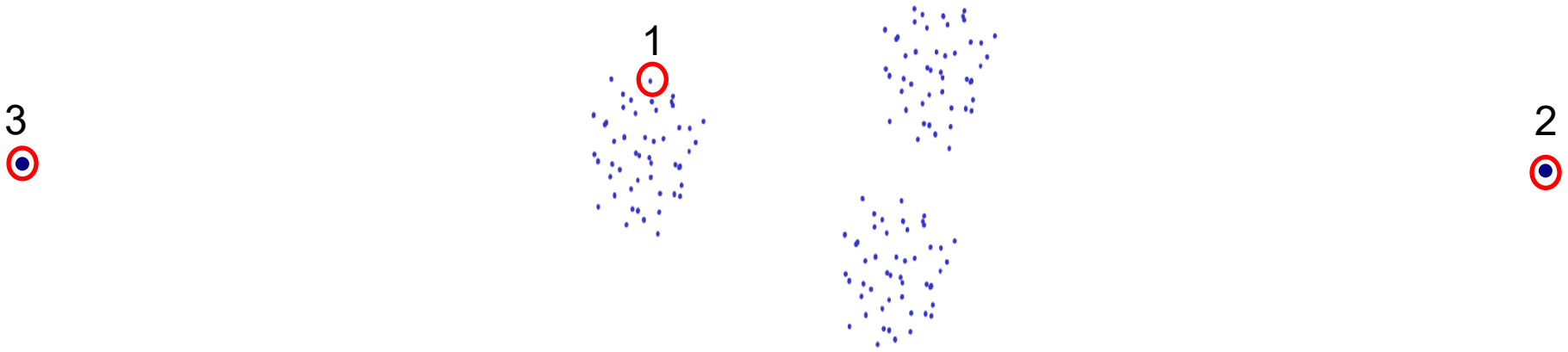
Furthest point heuristic:

- Pick first cluster center to be one arbitrarily selected point
- Iteratively pick cluster centers to be remaining points that are furthest from any selected point

Approximation algorithm to k-centers problem – minimize maximum distance of any point to its closest center

One idea – Furthest Point Heuristic

Question: Can this fail to lead to a good clustering?

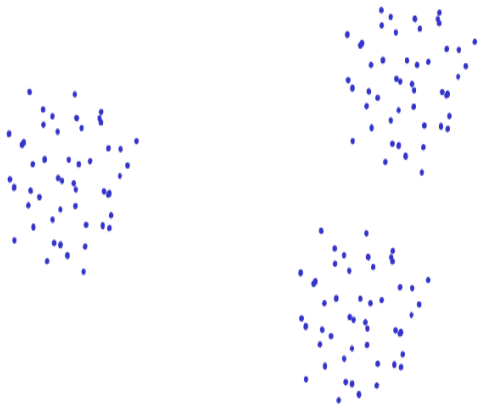


Furthest point heuristic:

- Pick first cluster center to be one arbitrarily selected point
- Iteratively pick cluster centers to be remaining points that are furthest from any selected point

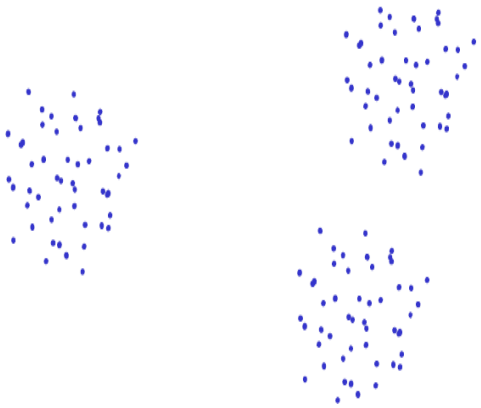
Approximation algorithm to k-centers problem – minimize maximum distance of any point to its closest center

Question: Any ideas for initializing the clusters that would spread the points while being less sensitive to outliers?



K-means++

Question: Any ideas for initializing the clusters that would spread the points while being less sensitive to outliers?



- Pick first cluster center to be one arbitrarily selected point
- Iteratively select a cluster center x' to be a point in the data probabilistically, where the probabilities are determined by

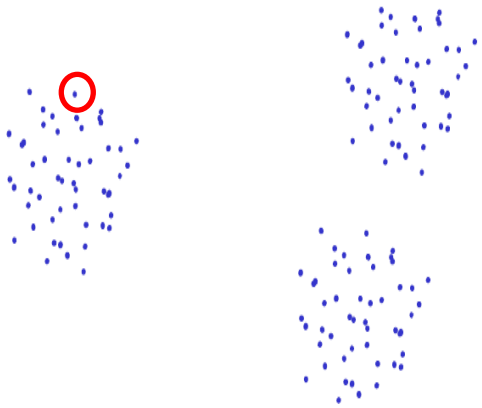
$$\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$$

where $D(x)$ is the distance of x to the closest selected center and X is the set of all points

Arthur and Vassilvitskii, SODA 2007

K-means++

Question: Any ideas for initializing the clusters that would spread the points while being less sensitive to outliers?



- Pick first cluster center to be one arbitrarily selected point
- Iteratively select a cluster center x' to be a point in the data probabilistically, where the probabilities are determined by

$$\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$$

where $D(x)$ is the distance of x to the closest selected center and X is the set of all points

Arthur and Vassilvitskii, SODA 2007

Run-time of k-means algorithm

Let *n* be the number of data points
Let *d* be the number of dimension of the data
Let *k* be the number of clusters
Let *i* be the number of iterations

What is the run-time of the k-means algorithm?

Run-time of k-means algorithm

Let *n* be the number of data points
Let *d* be the number of dimension of the data
Let *k* be the number of clusters
Let *i* be the number of iterations

What is the run-time of the k-means algorithm?

O(*ndki)*

Run-time of k-means algorithm

Let *n* be the number of data points
Let *d* be the number of dimension of the data
Let *k* be the number of clusters
Let *i* be the number of iterations

What is the run-time of the k-means algorithm?

$O(ndki)$

Number of iterations *i* worst case if run all the way to convergence is

$O(2^{\Omega(n^{.5})})$

in practice will need fewer iterations and can terminate early based on a fixed number of iterations or small changes to objective

Arthur and Vassilvitskii, SODA 2006