Clustering

Alan Moses

ml4bio

We want to find groups in data

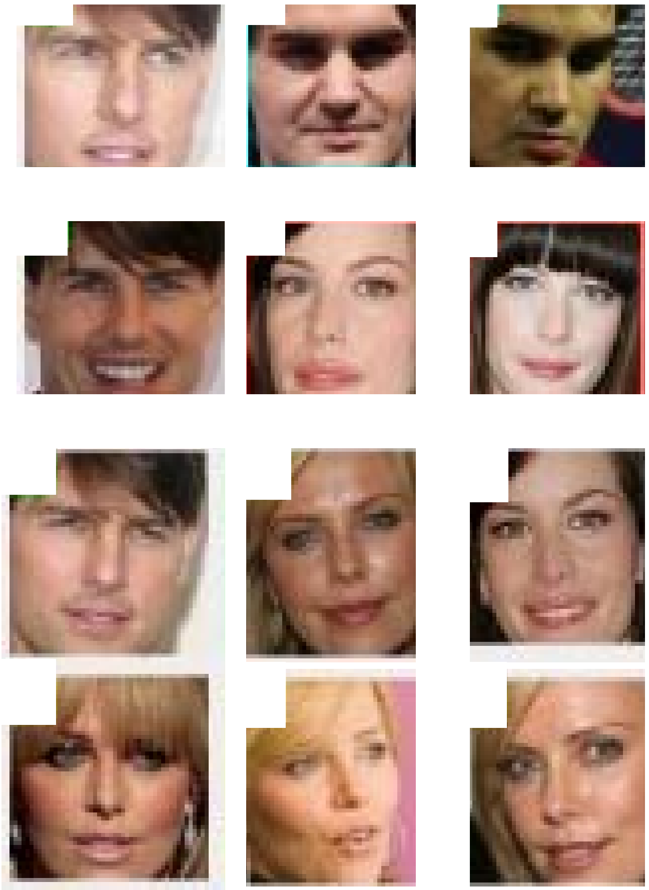
x_1=6.8

x_2=5.1

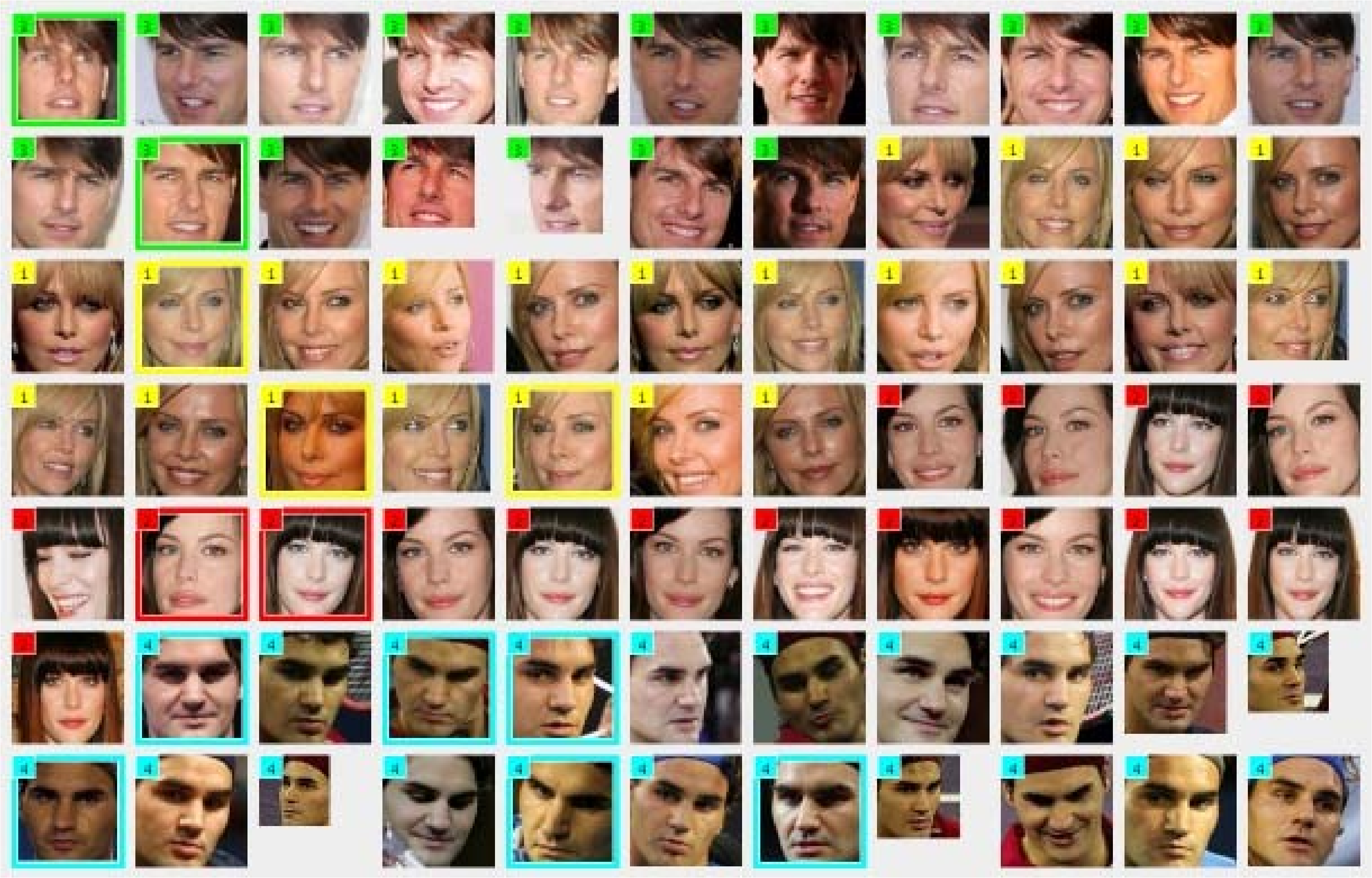
x_3=5.3

x_4=7.1

- How many groups are there?
- What data points (observations) belong to each group?



Ulrich Paquet



Ulrich Paquet

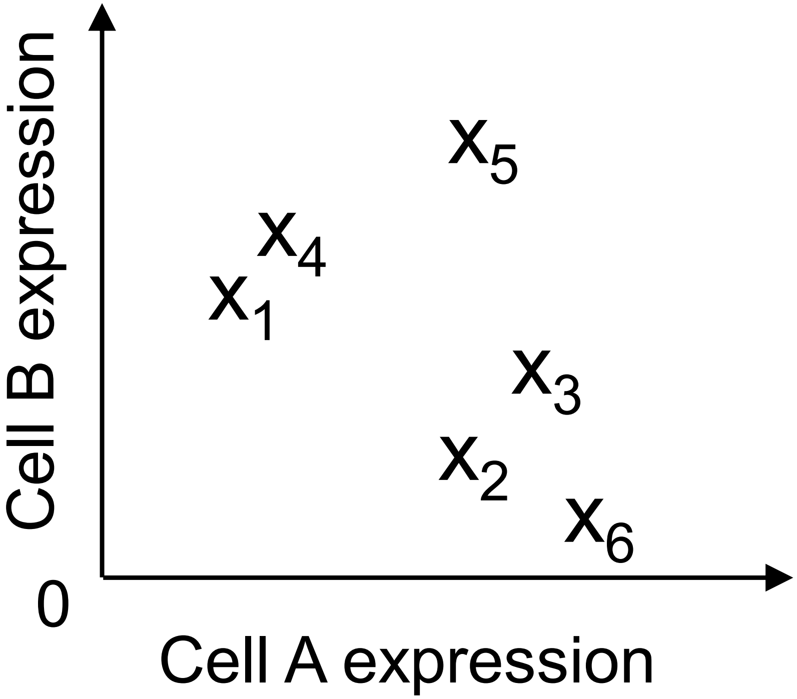
Outline

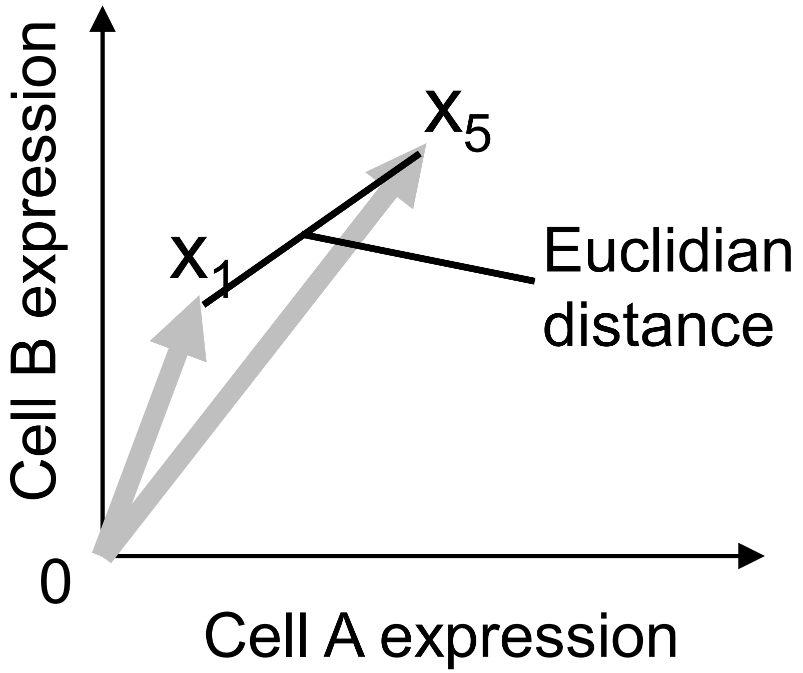
- Distance-based clustering
- Evaluating clustering
- How to choose the number of clusters

Distance between data points

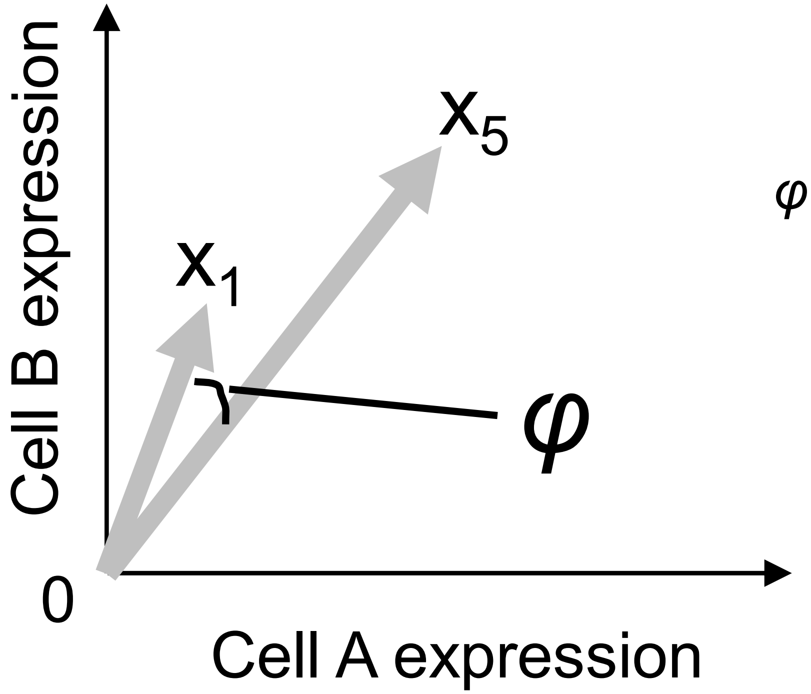
- In high dimensions there are different ways of defining a "distance"

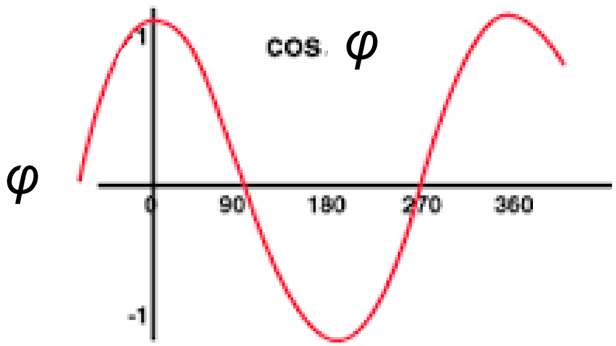
- For high-dimensional biology, the secret sauce is the weighting of dimensions


Cell B expression
x_5
x_4
x_1
x_3
x_2
x_6
0
Cell A expression



- This is the most intuitive distance to us…



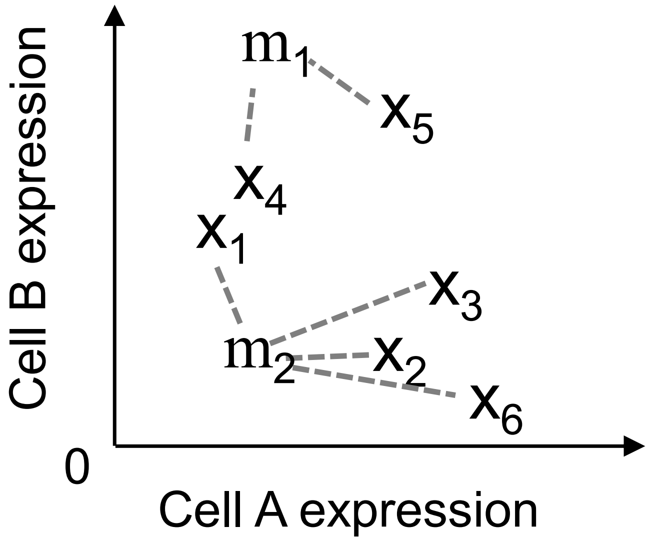


Use 1 - cos φ
as a distance

- Why might the angle be more reliable?

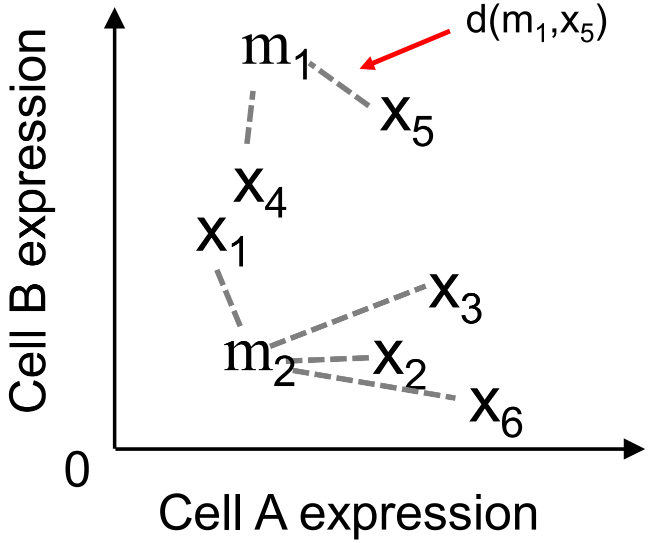
K-means clustering

- Statisticians sometimes call it c-means
- Traditionally uses the Euclidean distance
- Very intuitive objective function: Sum of squared distances (or errors) between datapoints and the closest "cluster mean".



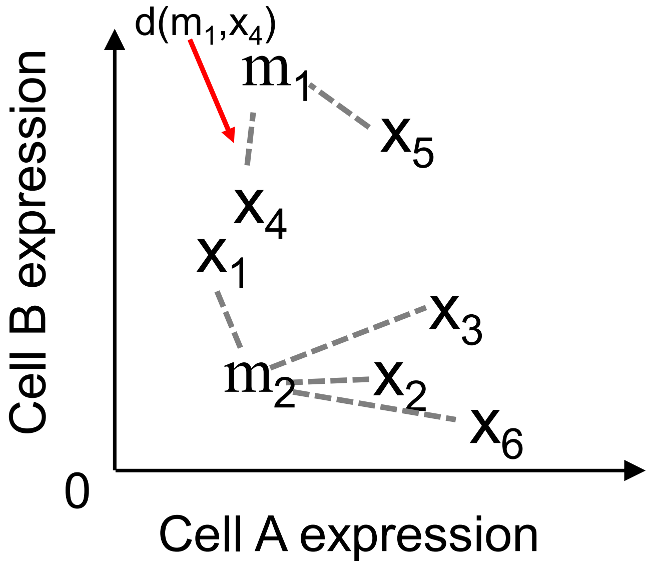
K-means clustering

- Statisticians sometimes call it c-means
- Traditionally uses the Euclidean distance
- Very intuitive objective function: Sum of squared distances (or errors) between datapoints and the closest "cluster mean".



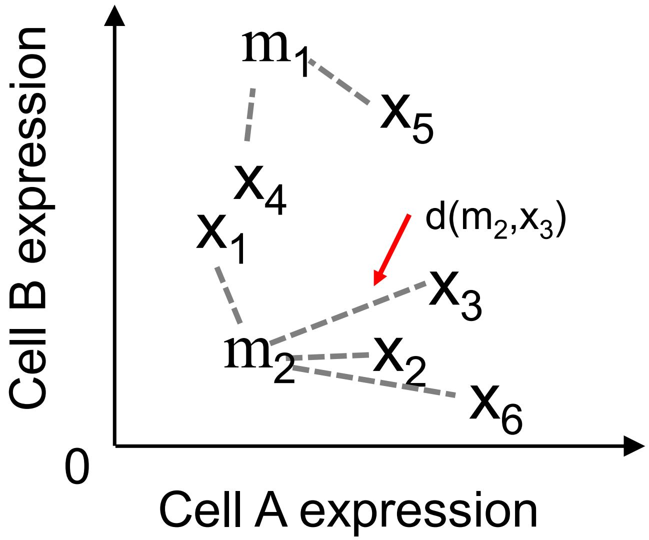
K-means clustering

- Statisticians sometimes call it c-means
- Traditionally uses the Euclidean distance
- Very intuitive objective function: Sum of squared distances (or errors) between datapoints and the closest "cluster mean".



K-means clustering

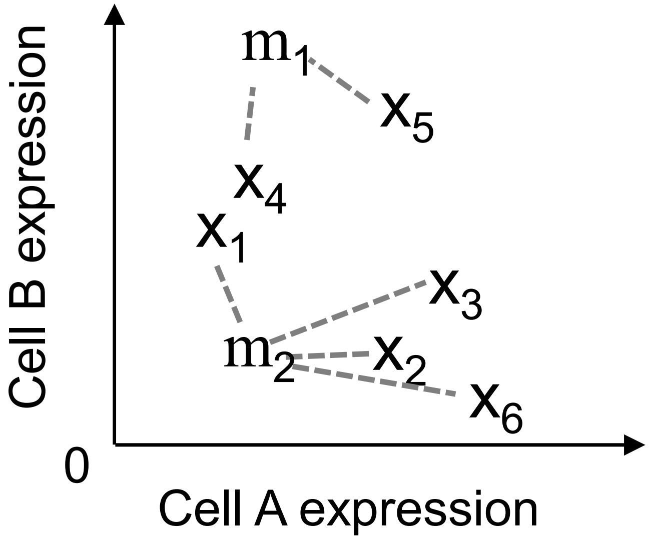
- Statisticians sometimes call it c-means
- Traditionally uses the Euclidean distance
- Very intuitive objective function: Sum of squared distances (or errors) between datapoints and the closest “cluster mean”.



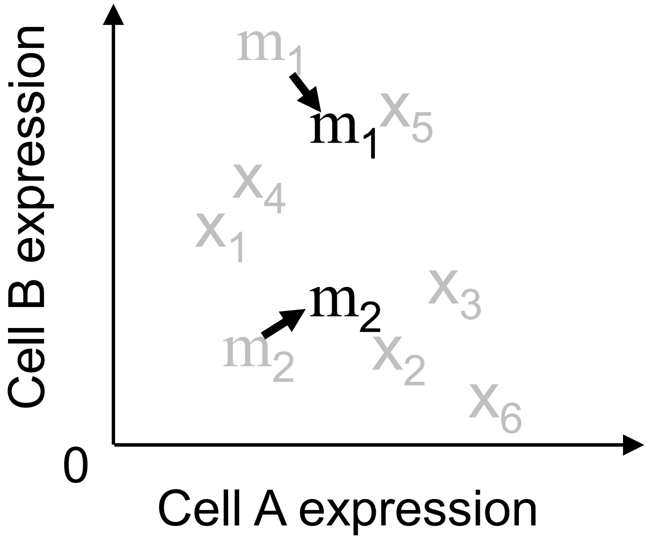
What are the parameters of this objective function?

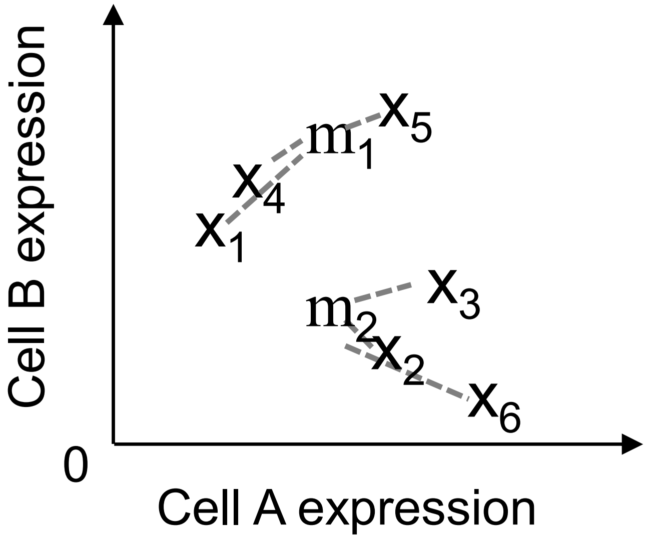
How do we chose the cluster means?

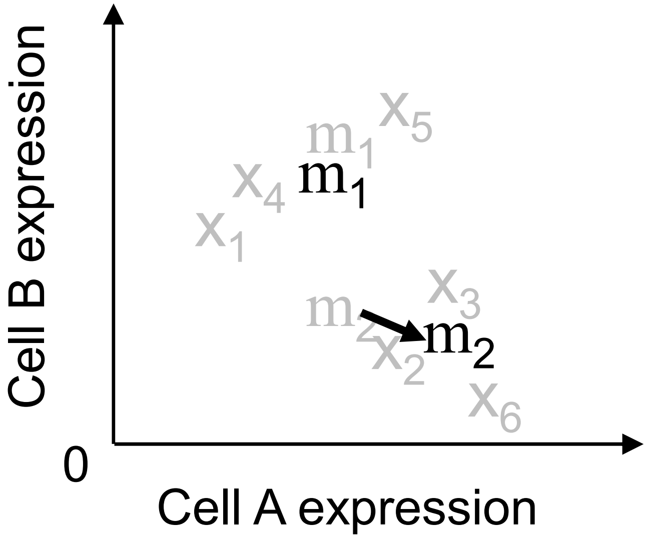
- In other words, how do we optimize the objective function?
    1. Start with k random cluster means
    2. Assign each datapoint to the closest cluster mean
    3. Recalculate the means so that it actually is the mean of the closest datapoints
    4. Compute the objective function
    5. Repeats steps 2-4 until the objective function doesn’t improve any more



Cell B expression
m_1
x_5
x_4
x_1
x_3
m_2
x_2
x_6
0
Cell A expression


Cell B expression
m_1
m_1
x_5
x_4
x_1
m_2
x_3
m_2
x_2
x_6
0
Cell A expression


Cell B expression
m1
x5
x4
x1
x3
m2
x2
x6
0
Cell A expression

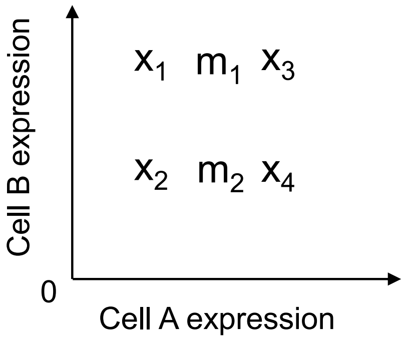


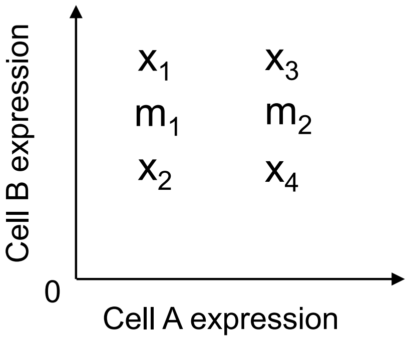
Machine learned!

How do we chose the cluster means?

- In other words, how do we optimize the objective function?
  - Turns out there is no analytic solution, and there are many local optima

k=2



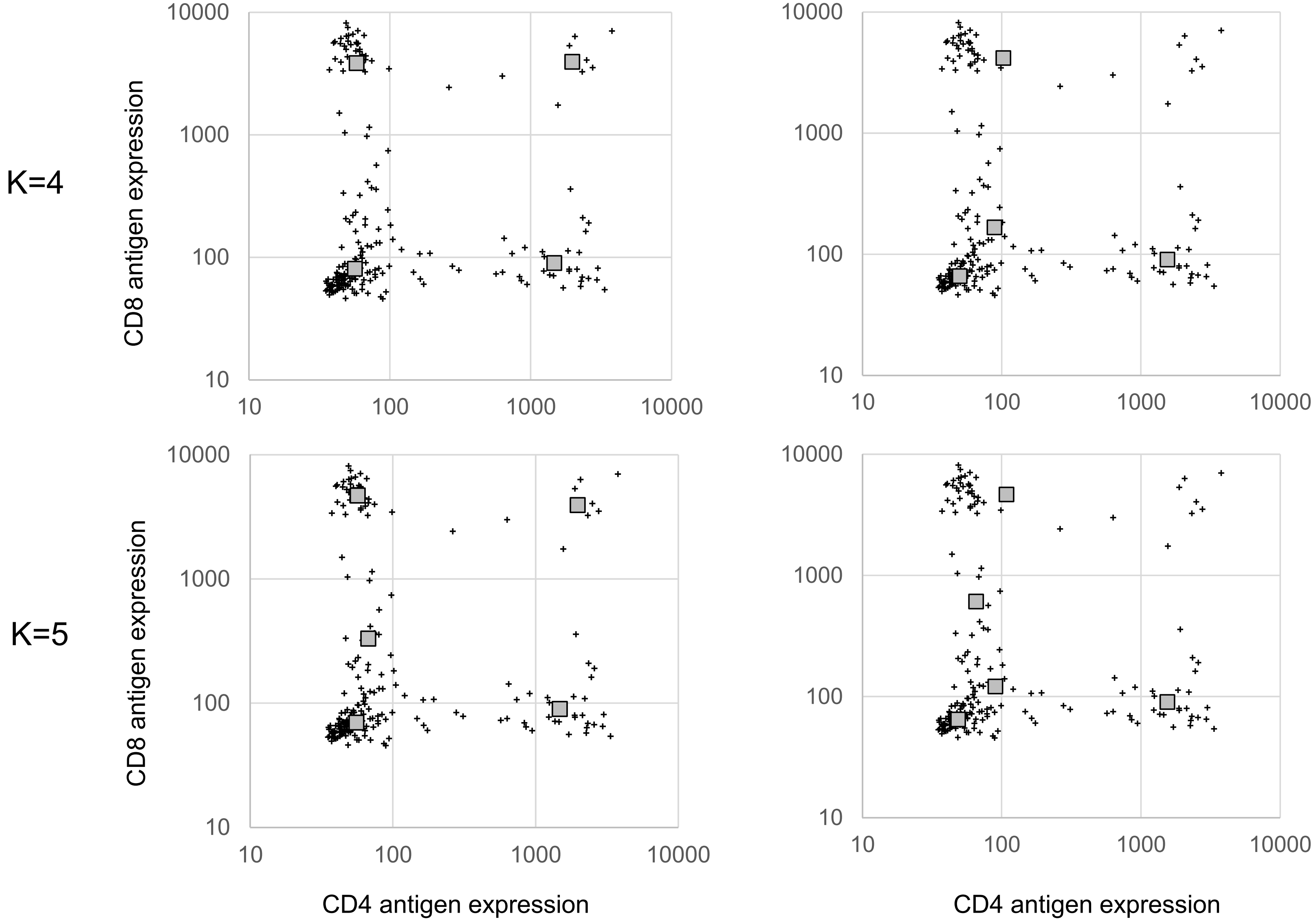


“The Quaid Data Box”

How do we chose the cluster means?

- In other words, how do we optimize the objective function?
  1. Start with k random cluster means
  2. Assign each datapoint to the closest cluster mean
  3. Recalculate the means so that it actually is the mean of the closest datapoints
  4. Compute the objective function
  5. Repeats steps 2-4 until the objective function doesn’t improve any more
  6. Repeat steps 1-5 a few more times (different random cluster means)

K-means is “stochastic”


K=4
K=5
CD8 antigen expression
CD8 antigen expression
10000
1000
100
10
10
100
1000
10000
CD4 antigen expression
CD4 antigen expression

Exemplar-based clustering

- K-medoids/partitioning around medoids (PAM)
- Can be better than k-means, but slower
- Newer exemplar-based methods are faster (e.g., affinity propagation)

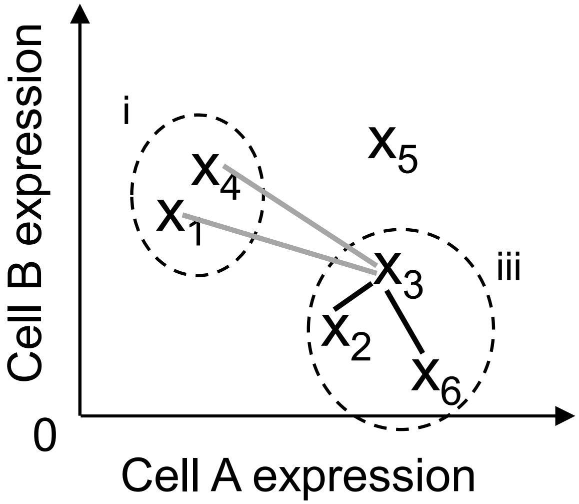
Instead of fiddling around with these “cluster means” choose an exemplar (a datapoint or observation) to represent each cluster

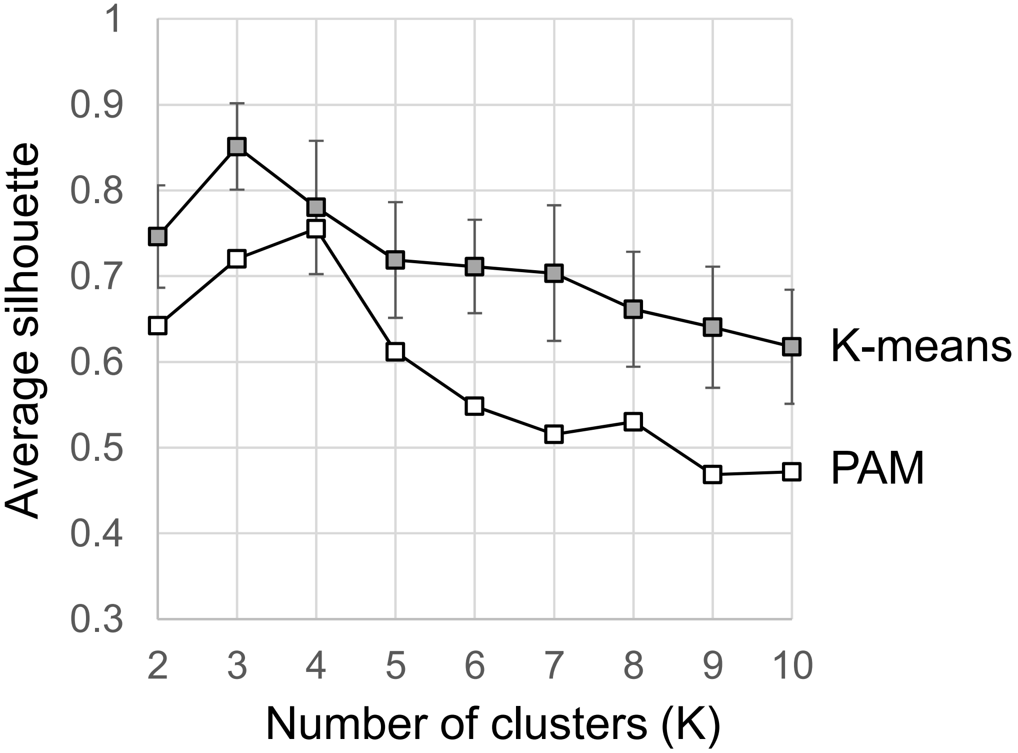
What is the objective function?
What are the parameters?

How to choose k?

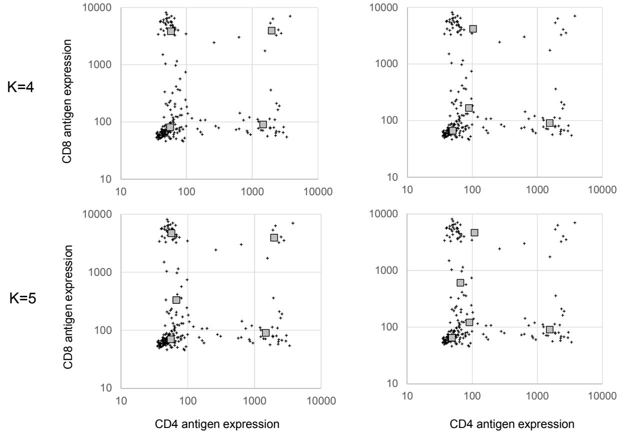
- Which distance/algorithm is better?

- Silhouette: compare the distances of points within the same clusters to the distances of points between clusters





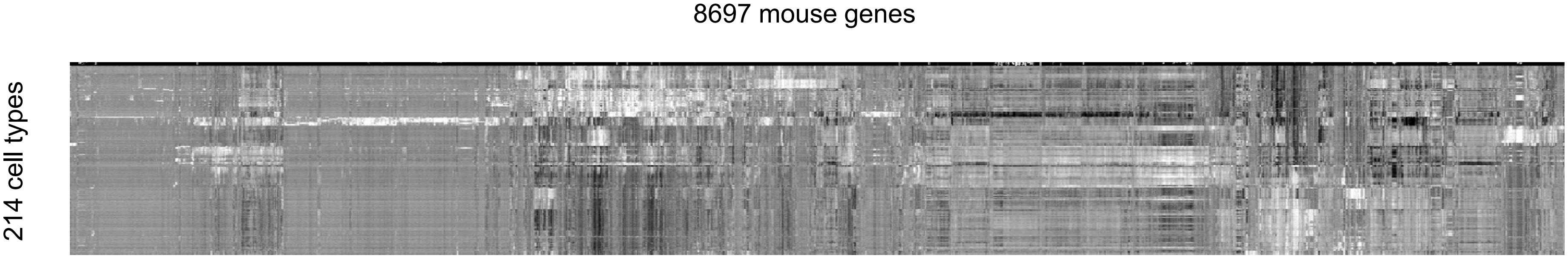
Average silhouette
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
2
3
4
5
6
7
8
9
10
Number of clusters (K)
K-means
PAM



K=4
K=5
CD8 antigen expression
CD8 antigen expression
CD4 antigen expression
CD4 antigen expression
10000
1000
100
10
10
100
1000
10000

None of these standard kinds of clustering work very well …

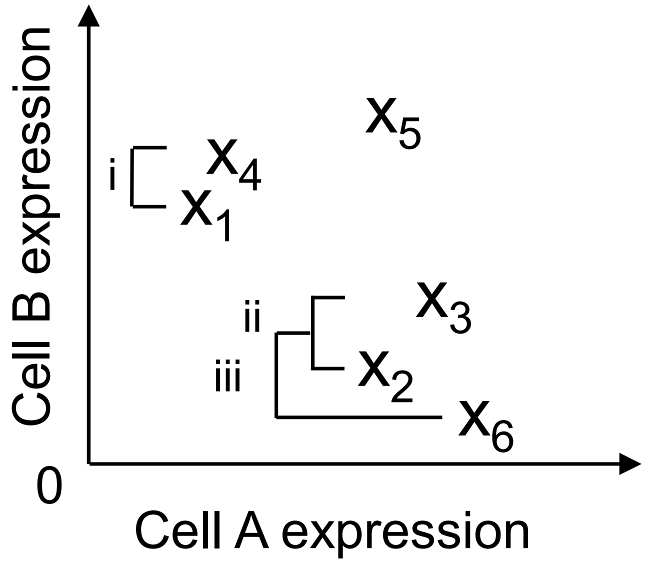
- If you have 100s of dimensions and many datapoints,
- k is probably large and unknown (or there really is no “best k”)



ImmGen data

Hierarchical clustering

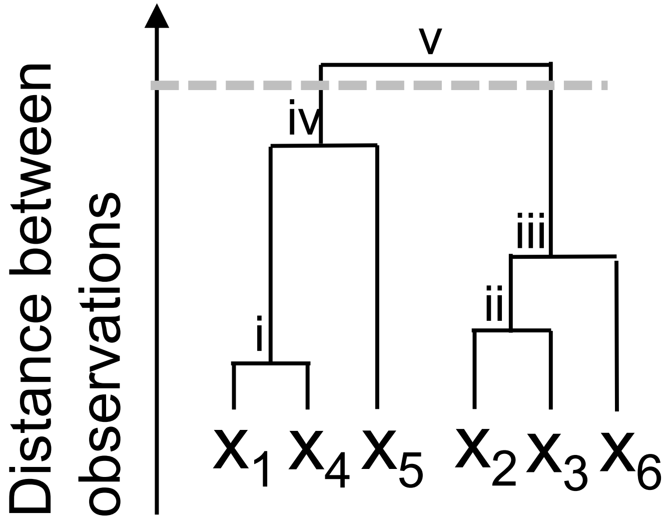
- Don’t bother with the k clusters



Successively group together closest datapoints, until all the data is joined together into a tree.

Hierarchical clustering

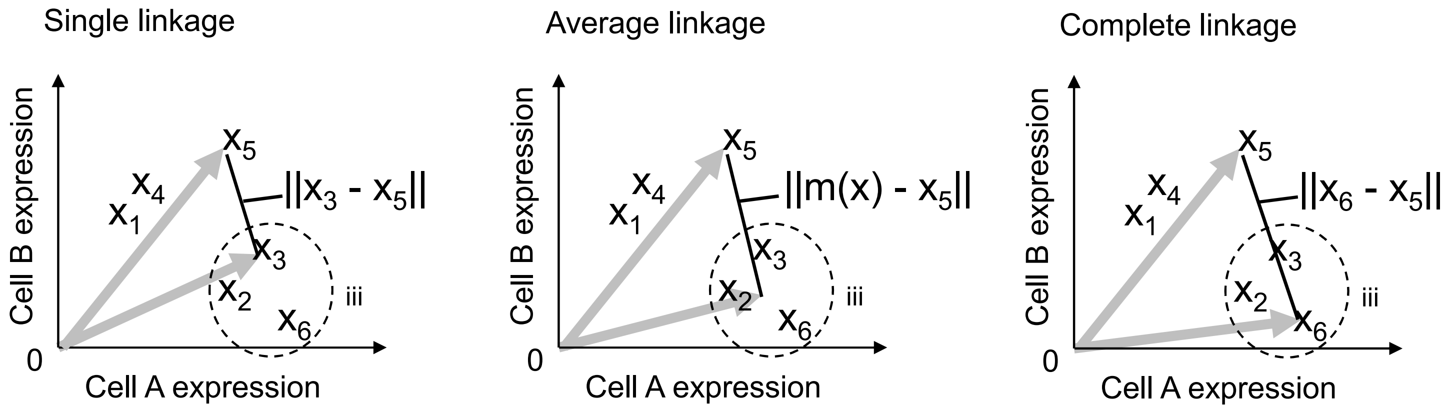
- Don’t bother with the k clusters



Cut the tree to find the clusters

Hierarchical clustering

- What is the distance between a point and a cluster?


Single linkage

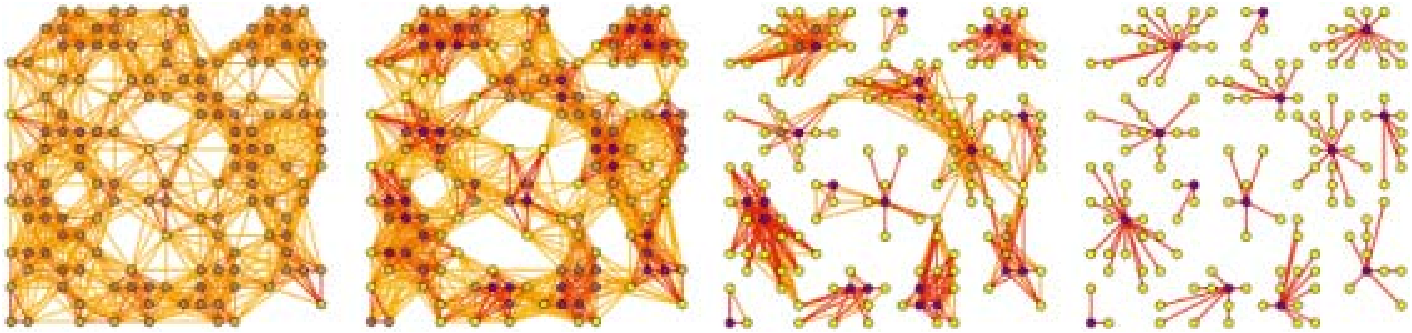
Average linkage

Complete linkage

UPGMA

Graph-based clustering

- No reason that the points have to be merged together into a tree
- All you need is a procedure to cut the graph

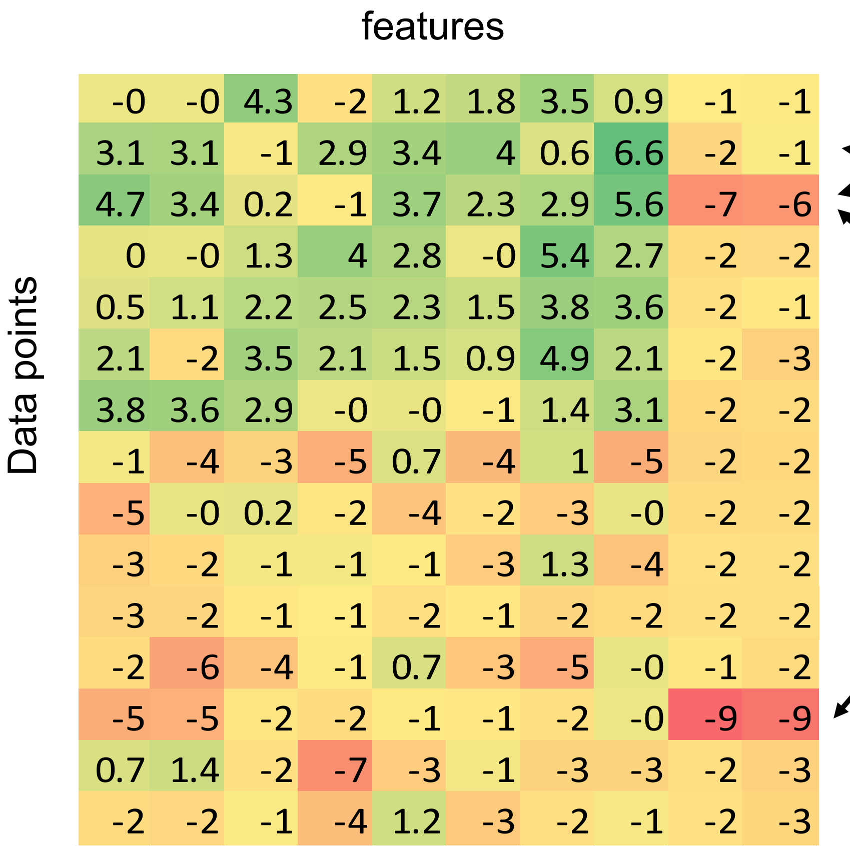


e.g., MCL (Markov Cluster Algorithm)

Secret sauce – weighting the dimensions

- For high-dimensional biological data, many of the dimensions might be similar
- If you cluster naively using Euclidean (or cosine) distance, you probably won’t get good clusters: the correlated dimensions will swamp out the signal from the uncorrelated dimensions
- You can try dimensionality reduction (but that’s a few lectures away)
- You can downweight the similar dimensions using a different type of distance
- Gene cluster 3.0 has a very good heuristic for this that works in high-dimensions

Fake data


features
Data points

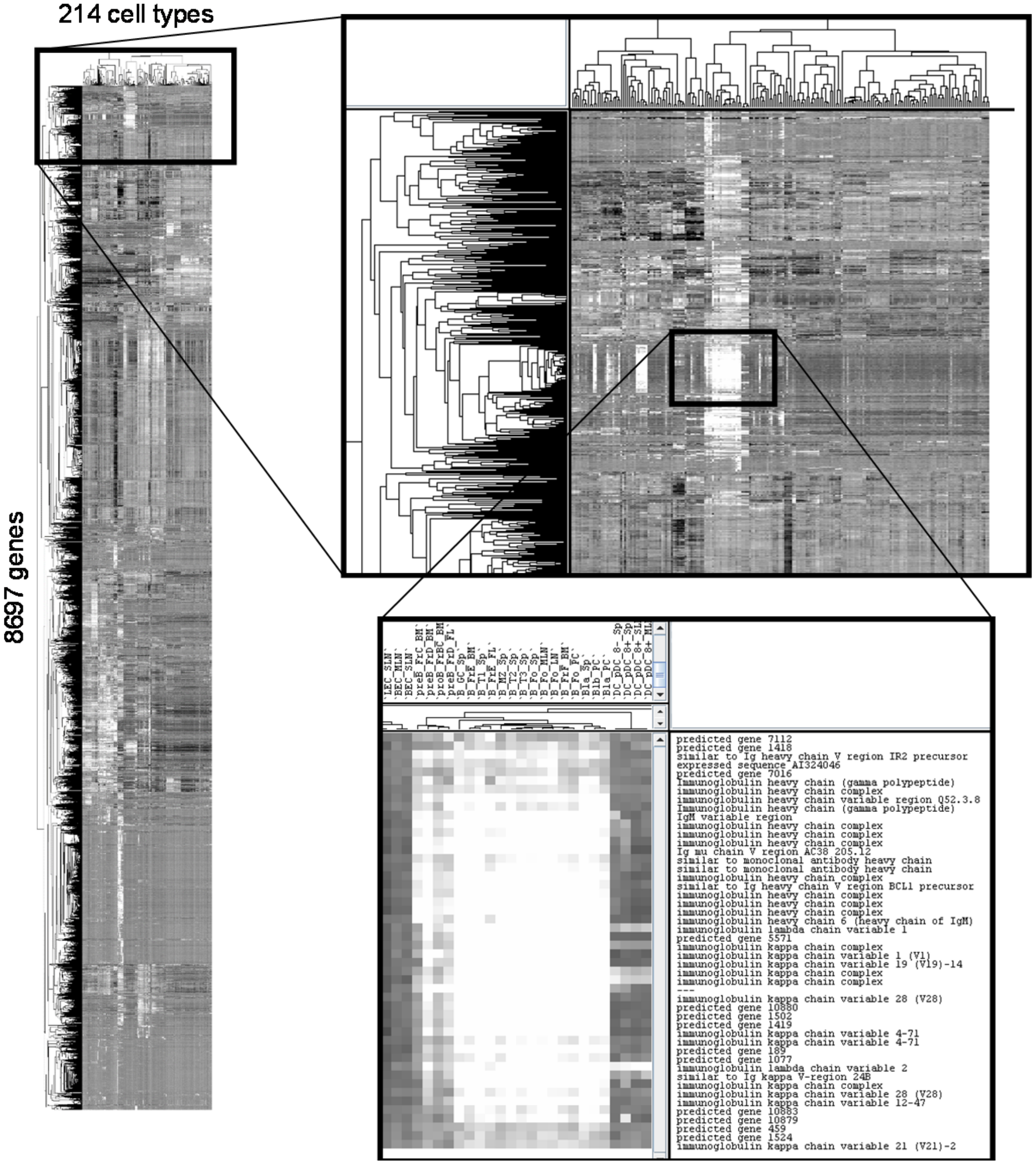
-0	-0	4.3	-2	1.2	1.8	3.5	0.9	-1	-1
3.1	3.1	-1	2.9	3.4	4	0.6	6.6	-2	-1
4.7	3.4	0.2	-1	3.7	2.3	2.9	5.6	-7	-6
0	-0	1.3	4	2.8	-0	5.4	2.7	-2	-2
0.5	1.1	2.2	2.5	2.3	1.5	3.8	3.6	-2	-1
2.1	-2	3.5	2.1	1.5	0.9	4.9	2.1	-2	-3
3.8	3.6	2.9	-0	-0	-1	1.4	3.1	-2	-2
-1	-4	-3	-5	0.7	-4	1	-5	-2	-2
-5	-0	0.2	-2	-4	-2	-3	-0	-2	-2
-3	-2	-1	-1	-1	-3	1.3	-4	-2	-2
-3	-2	-1	-1	-2	-1	-2	-2	-2	-2
-2	-6	-4	-1	0.7	-3	-5	-0	-1	-2
-5	-5	-2	-2	-1	-1	-2	-0	-9	-9
0.7	1.4	-2	-7	-3	-1	-3	-3	-2	-3
-2	-2	-1	-4	1.2	-3	-2	-1	-2	-3

These observations look similar, but it's mostly due to fluctuations in the correlated features
(Euclidean distance=9)

These are the datapoints you really want to identify as similar
(Euclidean distance=16.5)

Correlation distance also makes the top two datapoints more similar

Because of correlated, noisy features, clustering might fail to identify two similar datapoints…



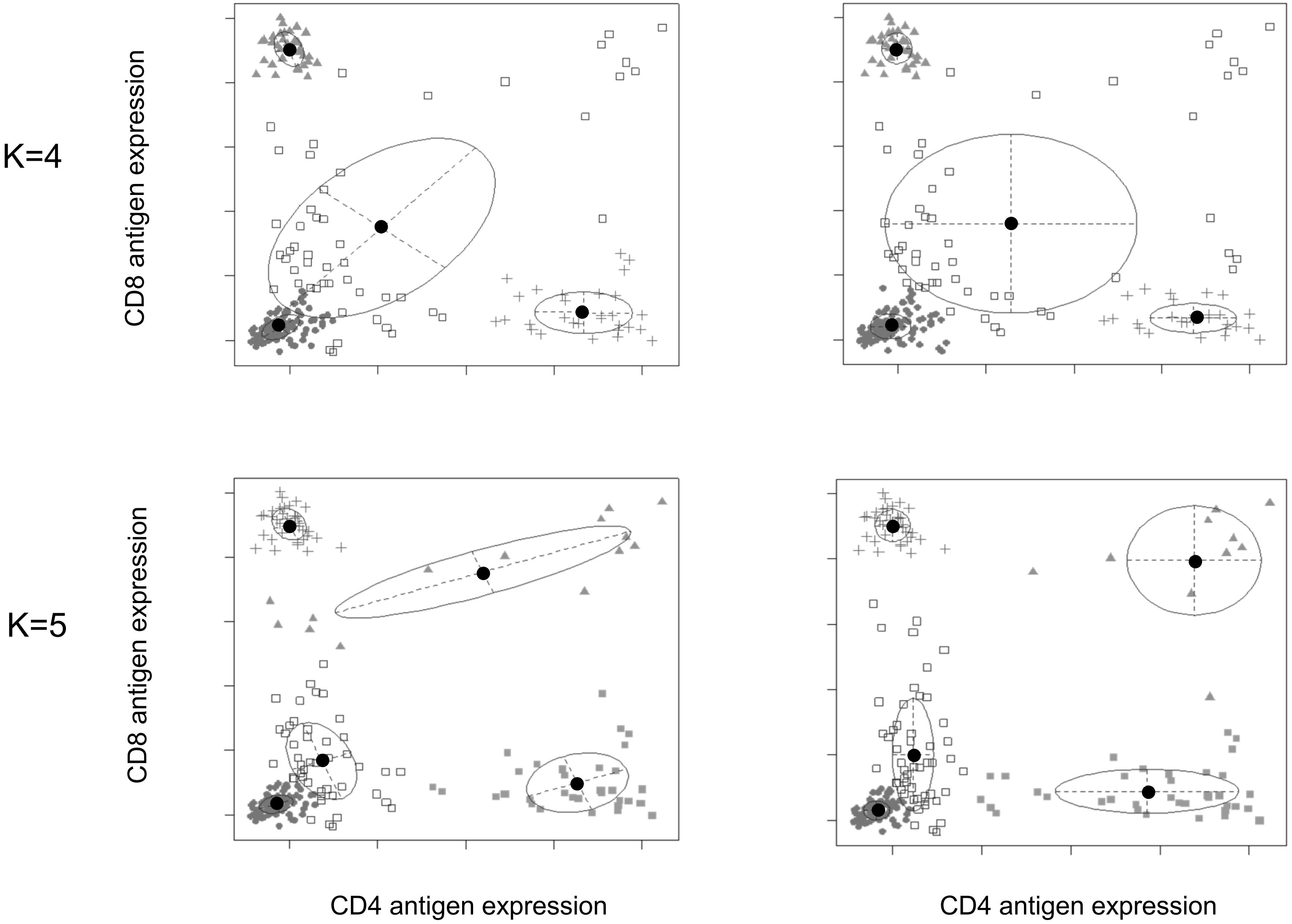
```
predicted gene 7112
predicted gene 1418
similar to Ig heavy chain V region IR2 precursor
expressed sequence AI324046
predicted gene 7016
Immunoglobulin heavy chain (gamma polypeptide)
immunoglobulin heavy chain complex
immunoglobulin heavy chain variable region Q52.3.8
Immunoglobulin heavy chain (gamma polypeptide)
IgM variable region
immunoglobulin heavy chain complex
immunoglobulin heavy chain complex
immunoglobulin heavy chain complex
Ig mu chain V region AC38 205.12
similar to monoclonal antibody heavy chain
similar to monoclonal antibody heavy chain
immunoglobulin heavy chain complex
similar to Ig heavy chain V region BCL1 precursor
immunoglobulin heavy chain complex
immunoglobulin heavy chain complex
immunoglobulin heavy chain complex
immunoglobulin heavy chain 6 (heavy chain of IgM)
immunoglobulin lambda chain variable 1
predicted gene 5571
immunoglobulin kappa chain complex
immunoglobulin kappa chain variable 1 (V1)
immunoglobulin kappa chain variable 19 (V19)-14
immunoglobulin kappa chain complex
immunoglobulin kappa chain complex
---
immunoglobulin kappa chain variable 28 (V28)
predicted gene 10880
predicted gene 1502
predicted gene 1419
immunoglobulin kappa chain variable 4-71
immunoglobulin kappa chain variable 4-71
predicted gene 189
predicted gene 1077
immunoglobulin lambda chain variable 2
similar to Ig kappa V-region 24B
immunoglobulin kappa chain complex
immunoglobulin kappa chain variable 28 (V28)
immunoglobulin kappa chain variable 12-47
predicted gene 10883
predicted gene 10879
predicted gene 459
predicted gene 1524
immunoglobulin kappa chain variable 21 (V21)-2
```

Hierarchical clustering of ImmGen data using gene cluster 3.0

Principled weighting - GMMs

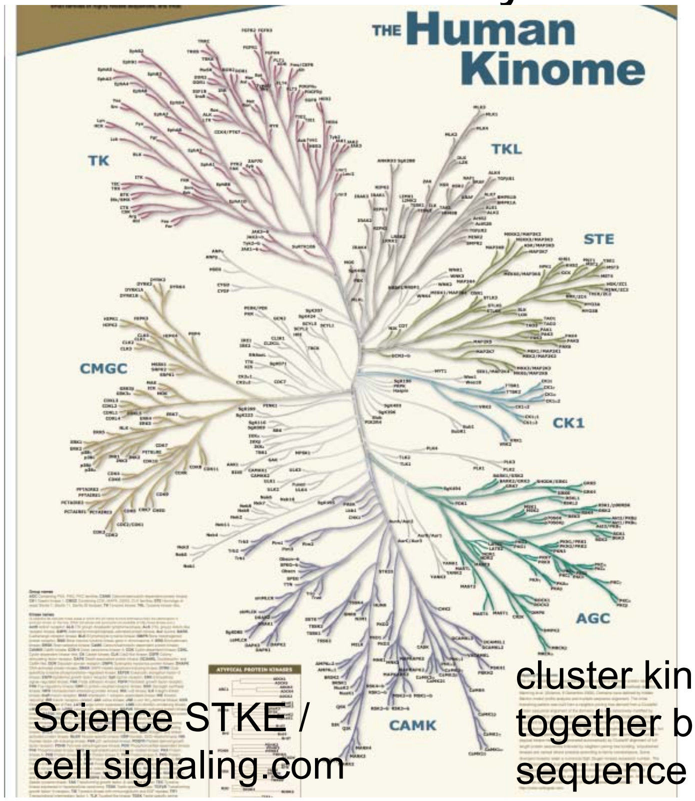
- Like K-means, but each cluster can have its own weighting! (locally warp space around each cluster mean)
- These “warping” parameters need to be learned from the data
- Optimization is still stochastic, called the “EM algorithm”
- Need regularization to make sure you aren’t overfitting – this can be done by subtracting a penalty proportional to the number of parameters. This is known as the AIC.

- Not really practical for high-dimensional clustering problems


K=4
K=5
CD8 antigen expression
CD8 antigen expression
CD4 antigen expression
CD4 antigen expression

Clustering other kinds of data

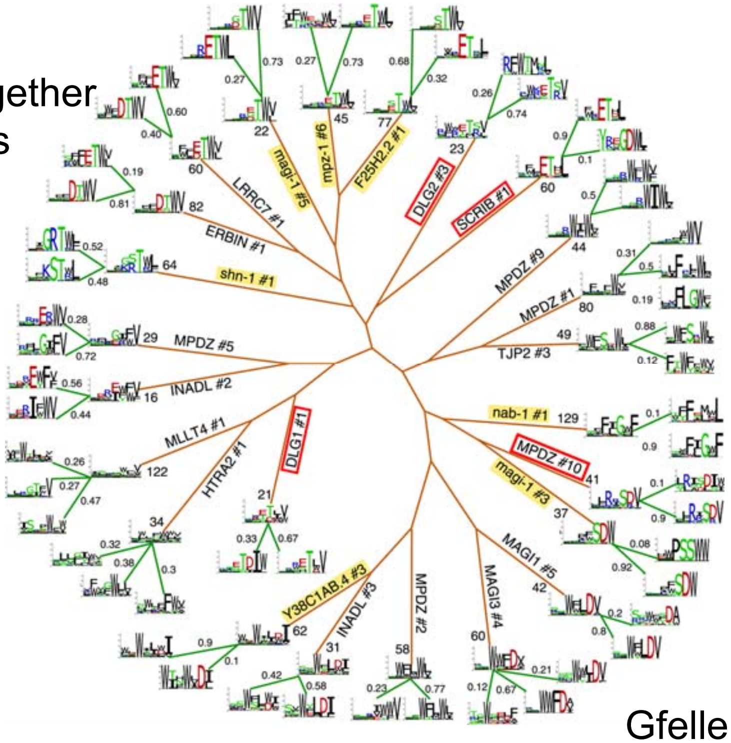
- Any observations where you can calculate distances can be clustered by k-means or UPGMA



Science STKE / cell signaling.com

cluster kinases together based on sequence similarity

Cluster together PDZ motifs



Gfeller *et al* 2011

The devil in in the distances

$$d(X_g, X_h) = \sqrt{\sum_{i=1}^{n} (X_{gi} - X_{hi})^2} = \sqrt{(X_g - X_h)^T (X_g - X_h)}$$ Euclidean distance

$$d(X_g, X_h) = \sqrt{(X_g - X_h)^T S^{-1} (X_g - X_h)}$$ Malhalanobis distance

$$d(X_g, X_h) = 1 - \cos\varphi(X_g, X_h) = 1 - \frac{{X_g}^T X_h}{||X_g||\ ||X_h||}$$ Cosine distance

Can any of these be applied to sequences?

How to represent sequences as numbers?

- Say our observation is the sequence *X*=CACGTG
- We can write a matrix

$$X = \underbrace{\begin{matrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{matrix}}_{L}$$

- For two sequences, X_1 and X_2, sum up the cosine distance for each position

$$\sum_{j=1}^{L}\left(1-\frac{{X_{1j}}^{T}X_{2j}}{||X_{1j}||\;||X_{2j}||}\right) = L - {X_1}^{T}X_2$$

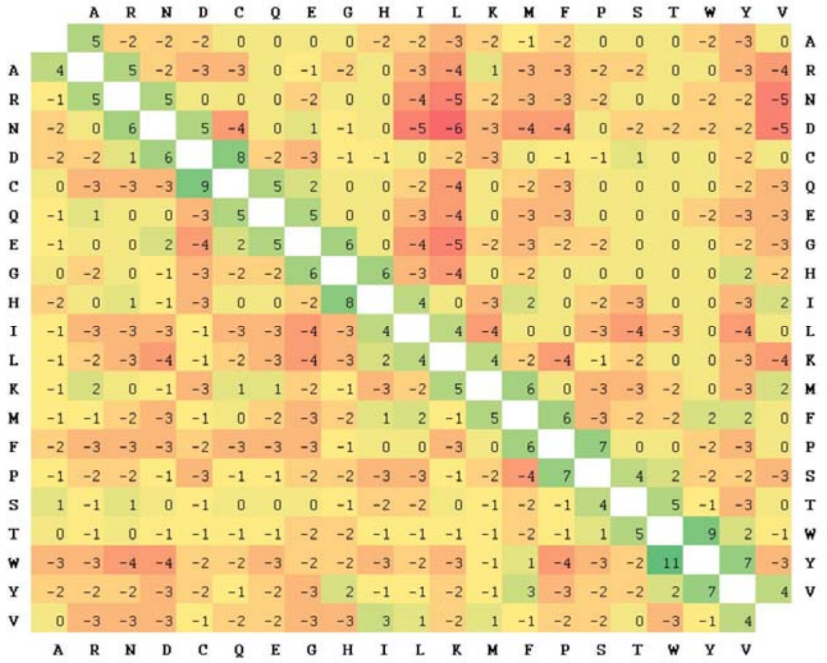
What does this distance measure?

What about weighting the dimensions?

- For sequences (especially proteins) different residues are counted differently

$$
\begin{aligned}
& d(X_1, X_2) \\
& = -\sum_{j=1}^{L} \sum_{a} \sum_{b} X_{1ja} M_{ab} X_{2jb} \\
& = -\sum_{j=1}^{L} {X_{1j}}^T M X_{2j} \\
& = -{X_1}^T M X_2
\end{aligned}
$$

BLOSUM62



		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	
		5	-2	-2	-2	0	0	0	0	-2	-2	-3	-2	-1	-2	0	0	0	-2	-3	0	A
A	4		5	-2	-3	-3	0	-1	-2	0	-3	-4	1	-3	-3	-2	-2	0	0	-3	-4	R
R	-1	5		5	0	0	0	-2	0	0	-4	-5	-2	-3	-3	-2	0	0	-2	-2	-5	N
N	-2	0	6		5	-4	0	1	-1	0	-5	-6	-3	-4	-4	0	-2	-2	-2	-2	-5	D
D	-2	-2	1	6		8	-2	-3	-1	-1	0	-2	-3	0	-1	-1	1	0	0	-2	0	C
C	0	-3	-3	-3	9		5	2	0	0	-2	-4	0	-2	-3	0	0	0	0	-2	-3	Q
Q	-1	1	0	0	-3	5		5	0	0	-3	-4	0	-3	-3	0	0	0	-2	-3	-3	E
E	-1	0	0	2	-4	2	5		6	0	-4	-5	-2	-3	-2	-2	0	0	0	-2	-3	G
G	0	-2	0	-1	-3	-2	-2	6		6	-3	-4	0	-2	0	0	0	0	0	2	-2	H
H	-2	0	1	-1	-3	0	0	-2	8		4	0	-3	2	0	-2	-3	0	0	-3	2	I
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4		4	-4	0	0	-3	-4	-3	0	-4	0	L
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4		4	-2	-4	-1	-2	0	0	-3	-4	K
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5		6	0	-3	-3	-2	0	-3	2	M
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5		6	-3	-2	-2	2	2	0	F
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6		7	0	0	-2	-3	0	P
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7		4	2	-2	-2	-3	S
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4		5	-1	-3	0	T
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5		9	2	-1	W
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		7	-3	Y
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7		4	V
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4		
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V		

CCF53P62