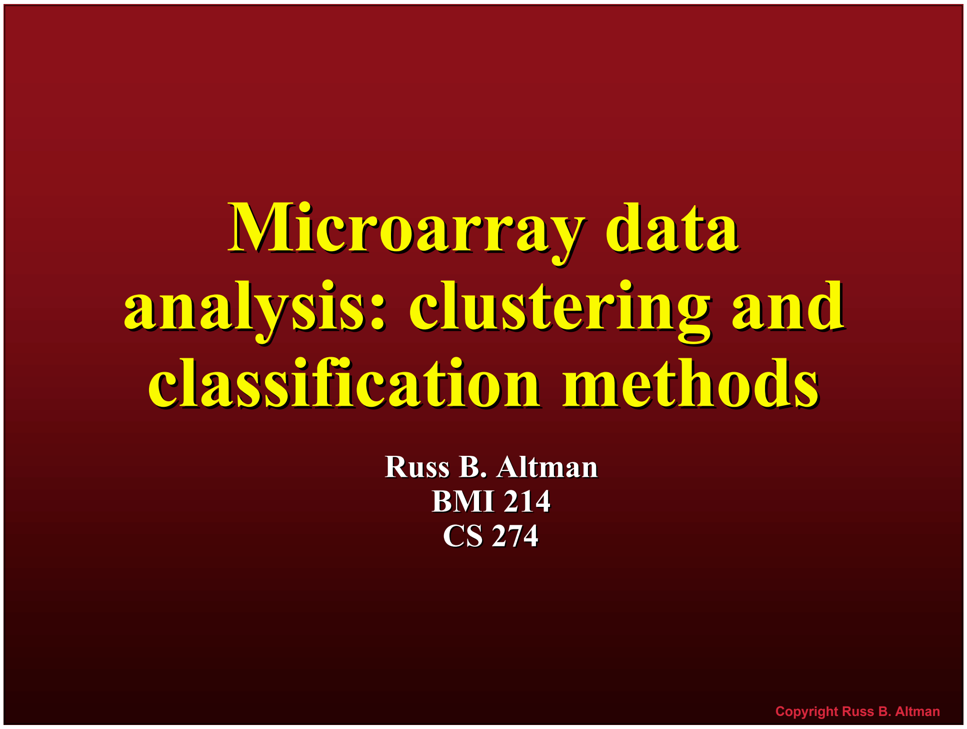
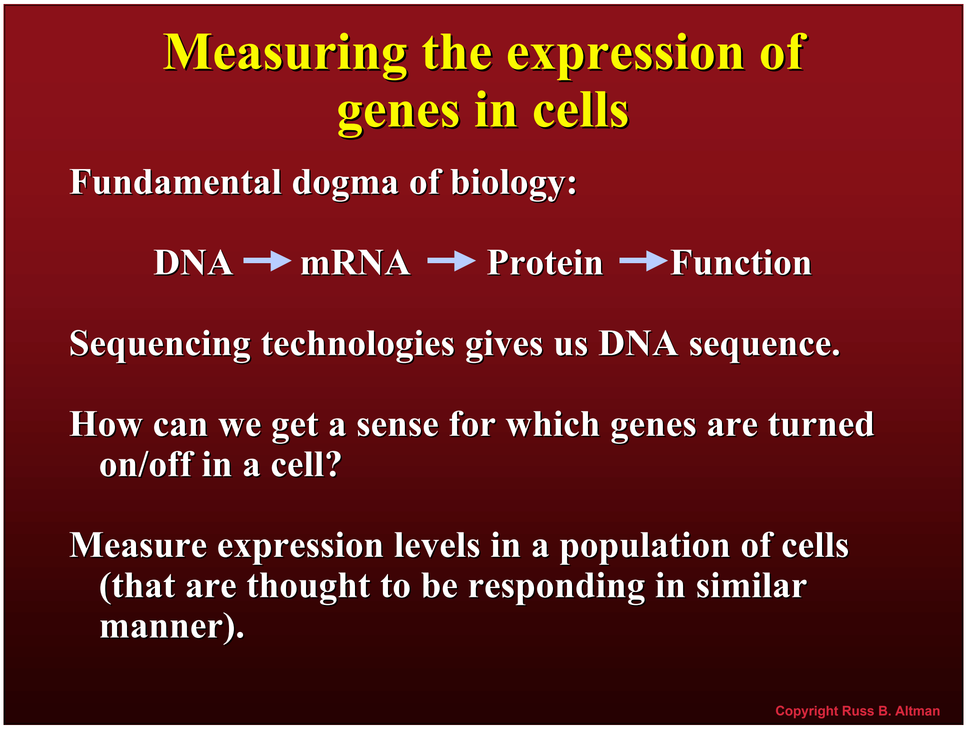


Microarray data analysis: clustering and classification methods

Russ B. Altman
BMI 214
CS 274



Measuring the expression of genes in cells

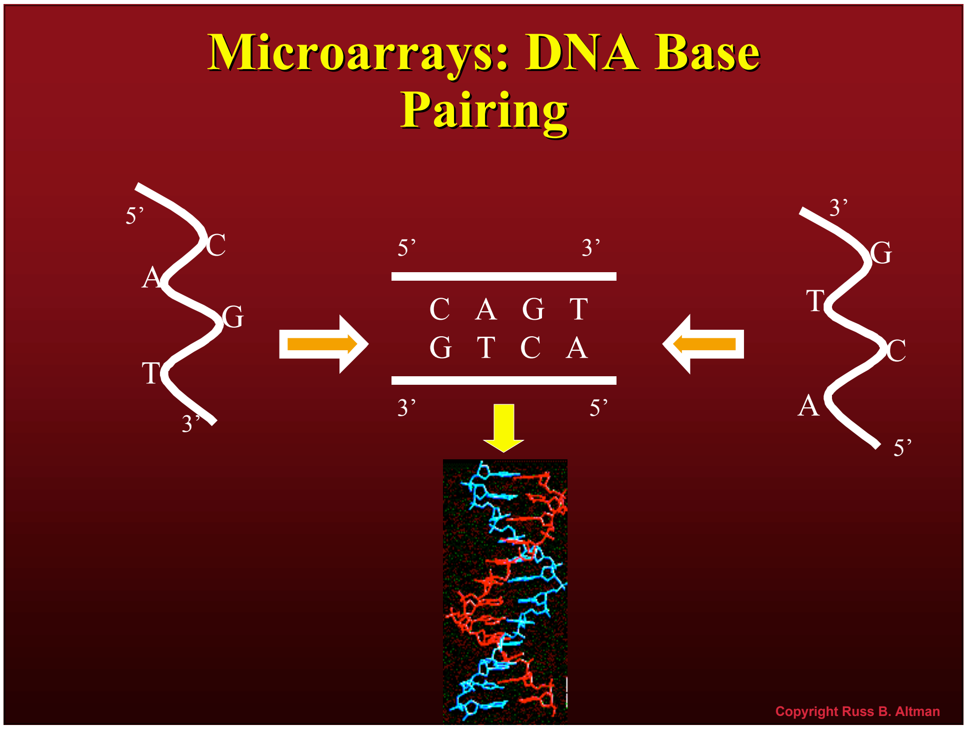
Fundamental dogma of biology:

DNA → mRNA → Protein → Function

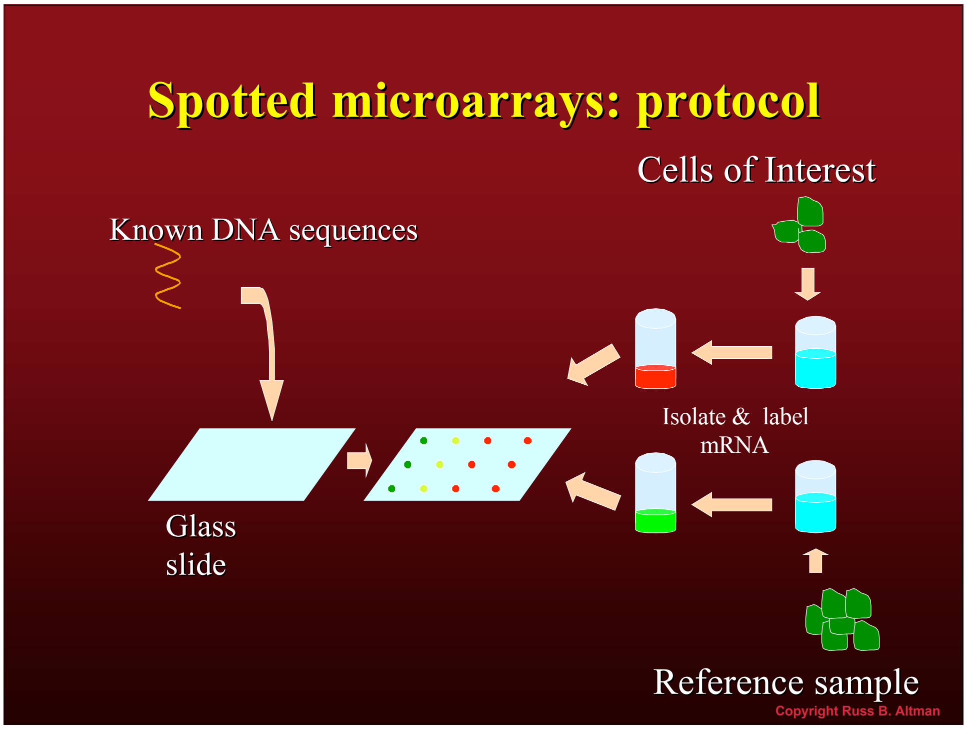
Sequencing technologies gives us DNA sequence.

How can we get a sense for which genes are turned on/off in a cell?

Measure expression levels in a population of cells (that are thought to be responding in similar manner).

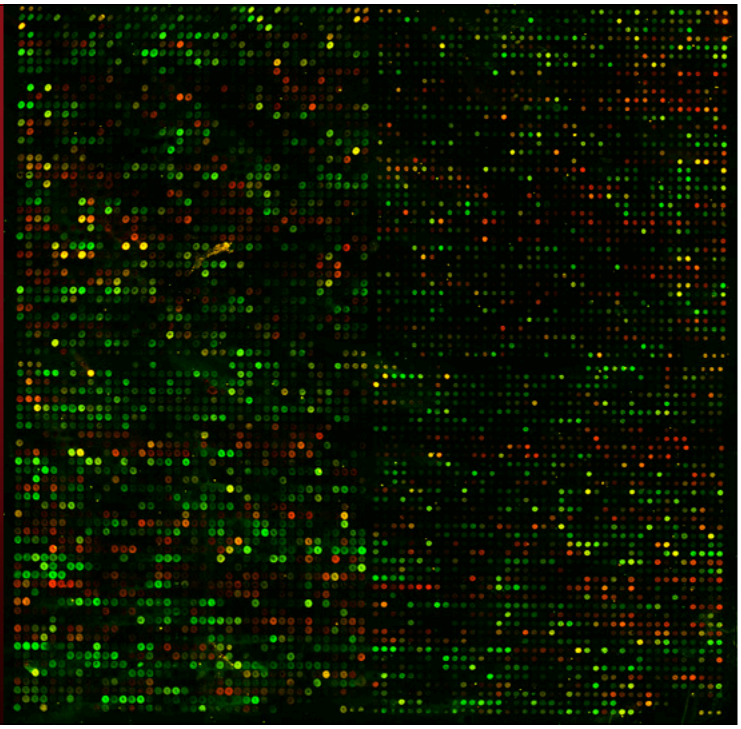


Microarrays: DNA Base Pairing



Spotted microarrays: protocol

Typical DNA array for Yeast

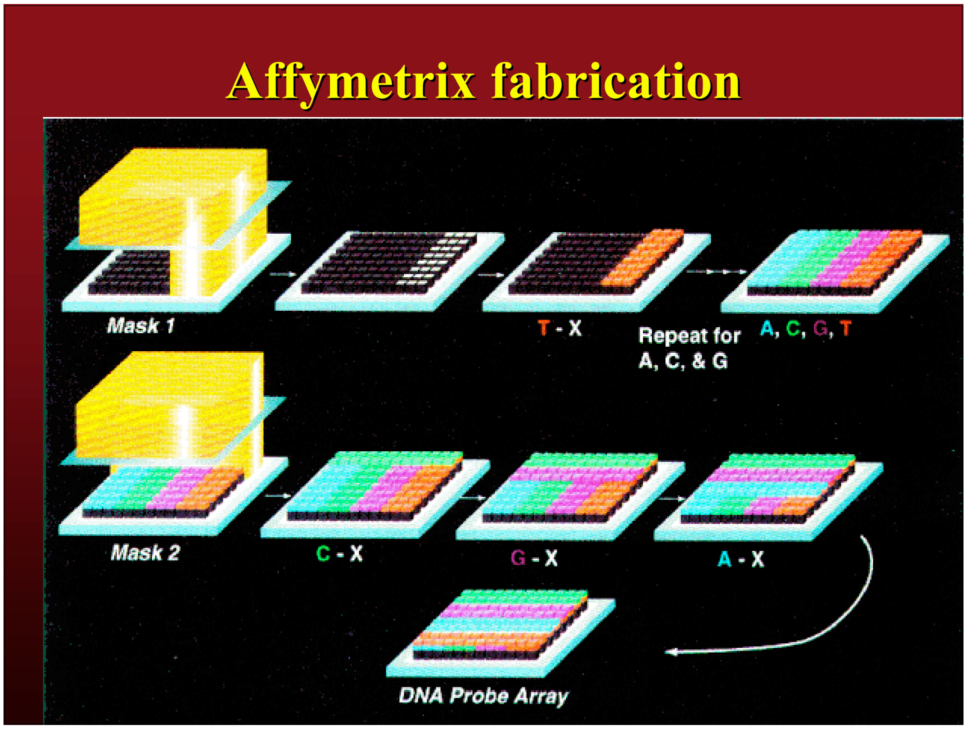


Affymetrix chip technology

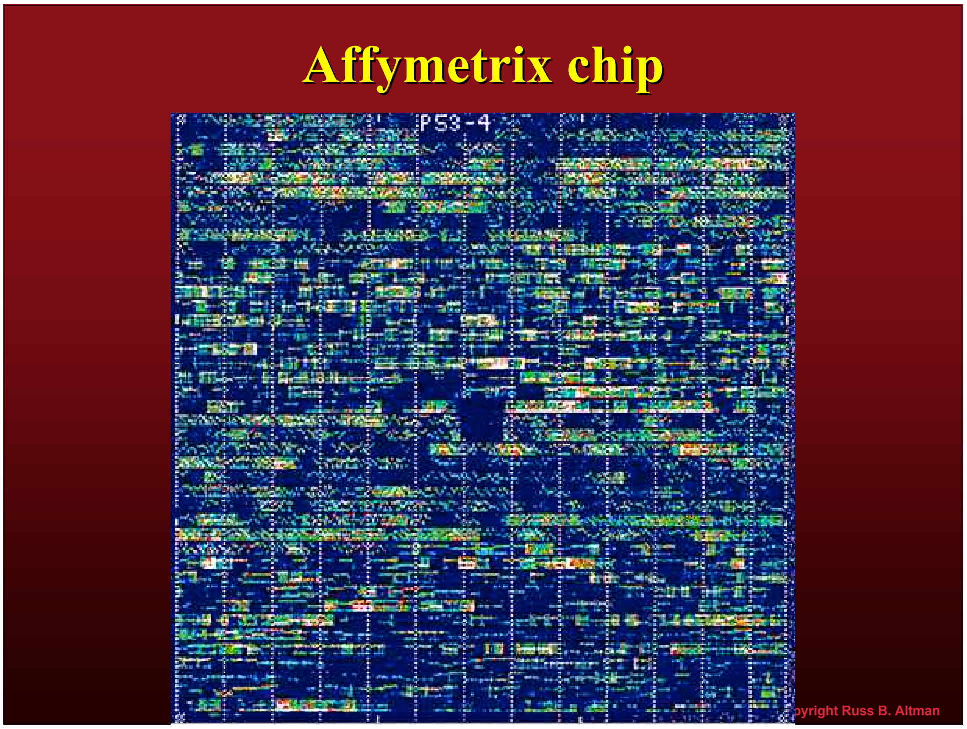
Instead of putting down intact genes on the chip, these chips put down N-mers of a certain length (around 20) systematically onto a chip by synthesizing the N-mers on the spots.

Labelled mRNA is then added to the chip and a *pattern* of binding (based on which 20-mers are in the mRNA sequence) is seen.

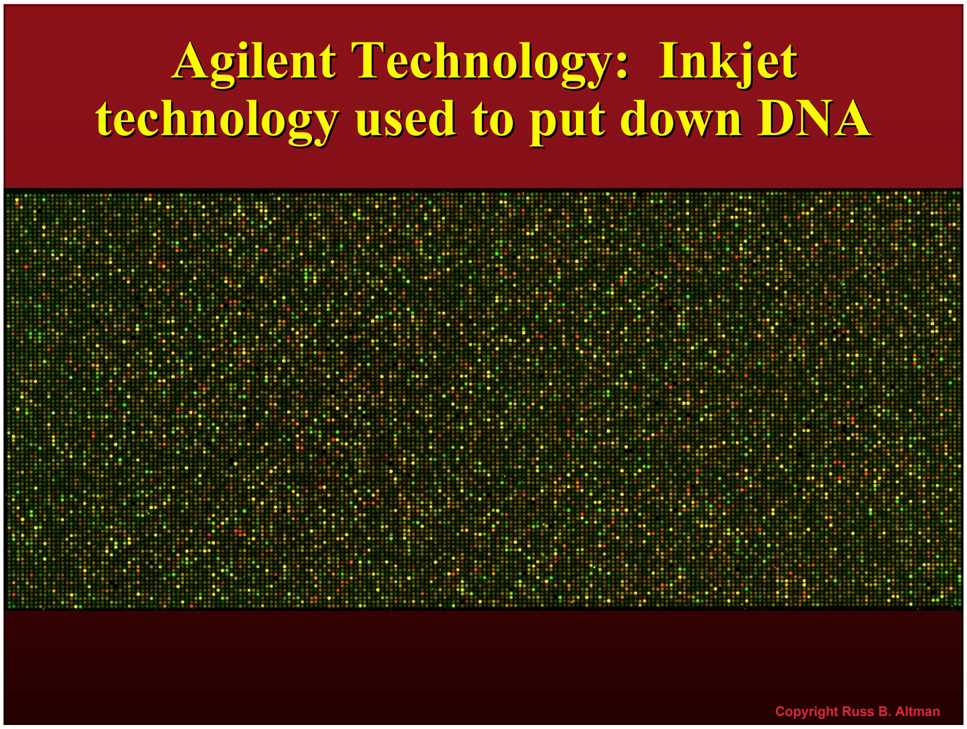
Bioinformatics is used to deduce the mRNA sequences that are present



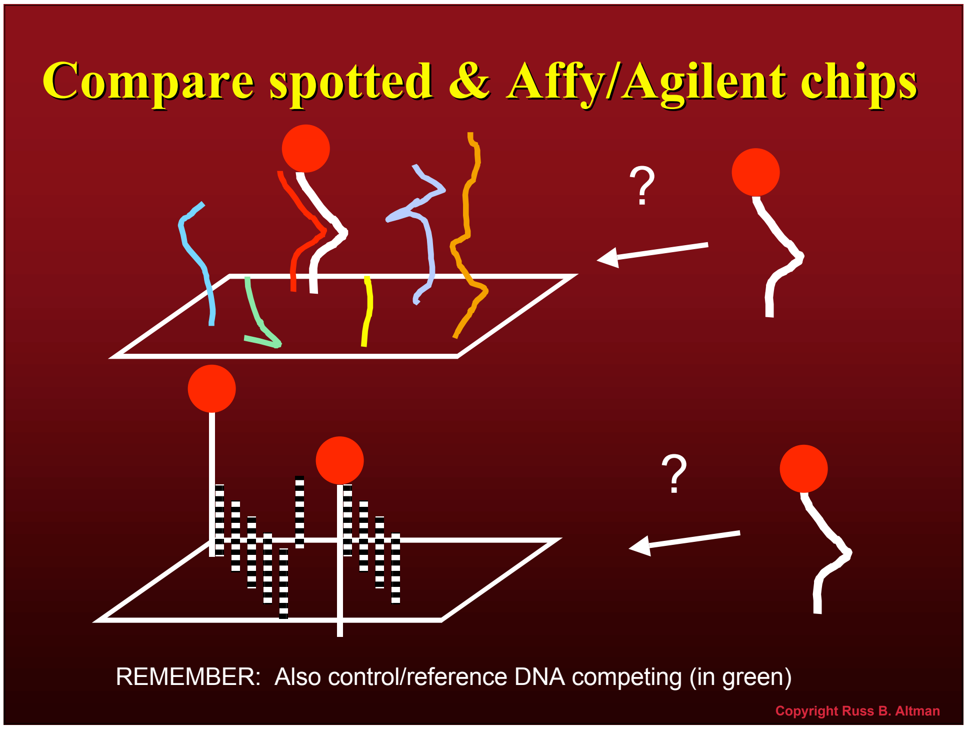
Affymetrix fabrication



Affymetrix chip

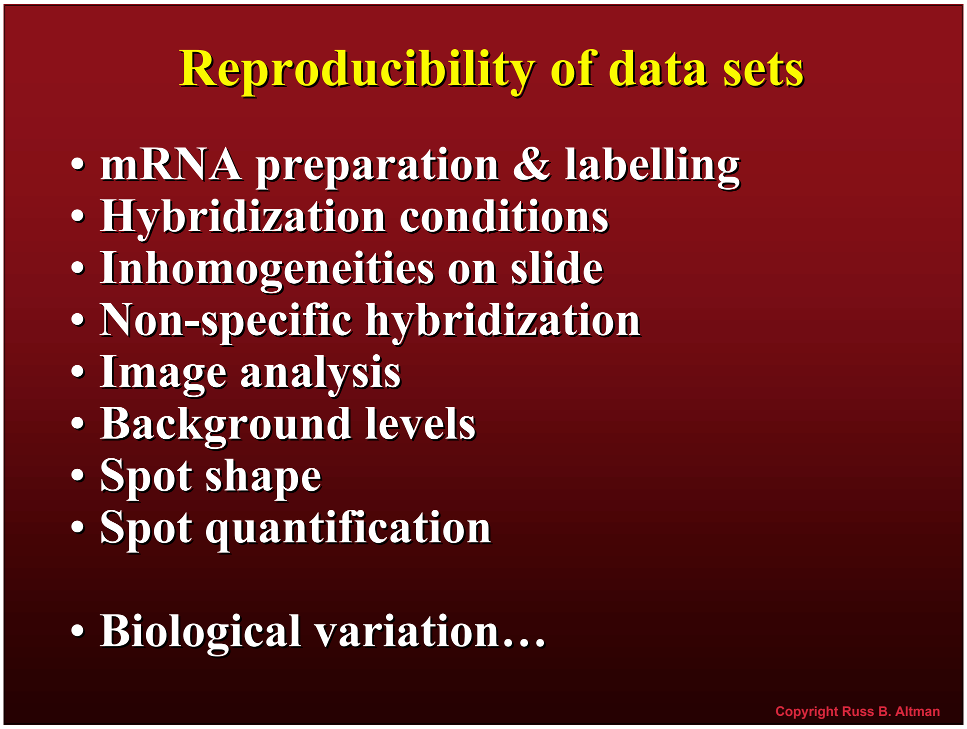


Agilent Technology: Inkjet technology used to put down DNA



Compare spotted & Affy/Agilent chips

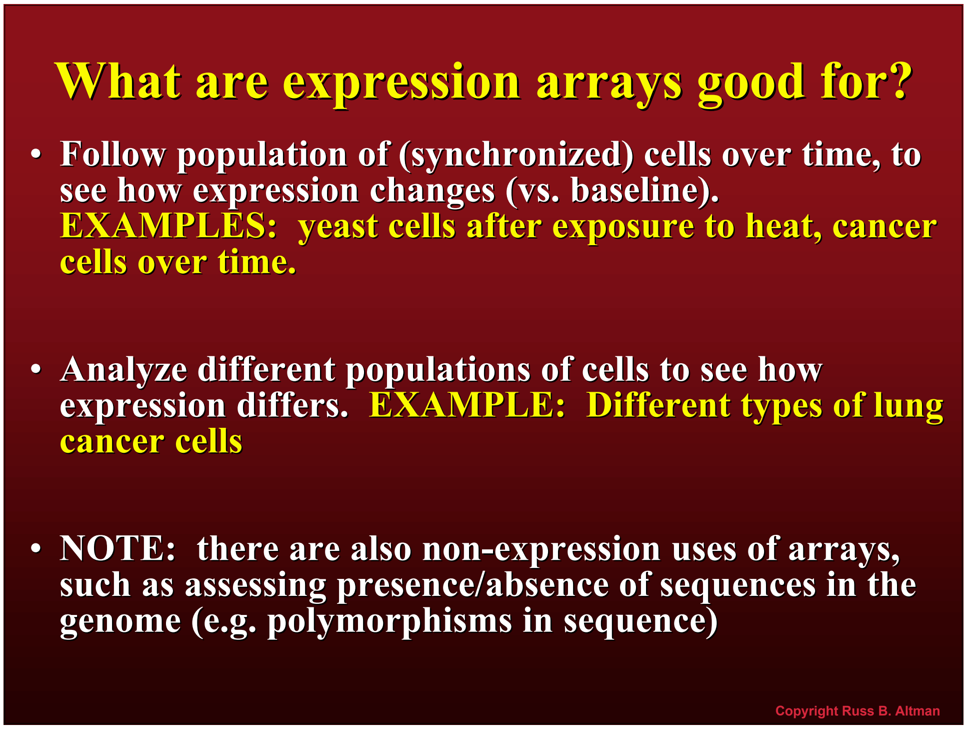
REMEMBER: Also control/reference DNA competing (in green)



Reproducibility of data sets

- mRNA preparation & labelling
- Hybridization conditions
- Inhomogeneities on slide
- Non-specific hybridization
- Image analysis
- Background levels
- Spot shape
- Spot quantification

- Biological variation…

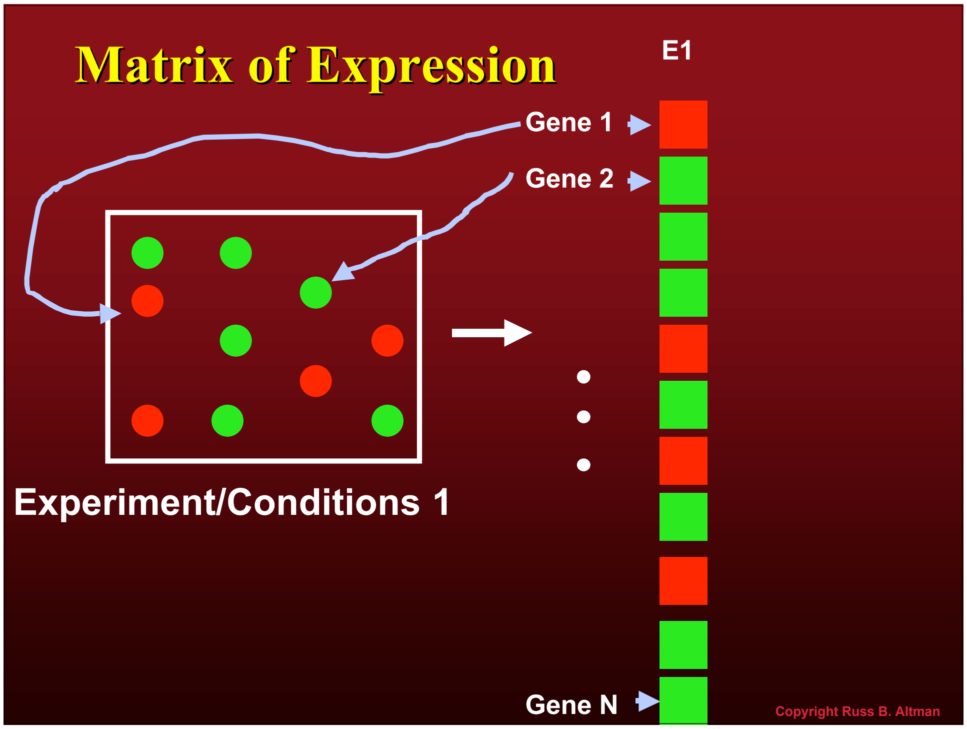


What are expression arrays good for?

- Follow population of (synchronized) cells over time, to see how expression changes (vs. baseline). EXAMPLES: yeast cells after exposure to heat, cancer cells over time.

- Analyze different populations of cells to see how expression differs. EXAMPLE: Different types of lung cancer cells

- NOTE: there are also non-expression uses of arrays, such as assessing presence/absence of sequences in the genome (e.g. polymorphisms in sequence)

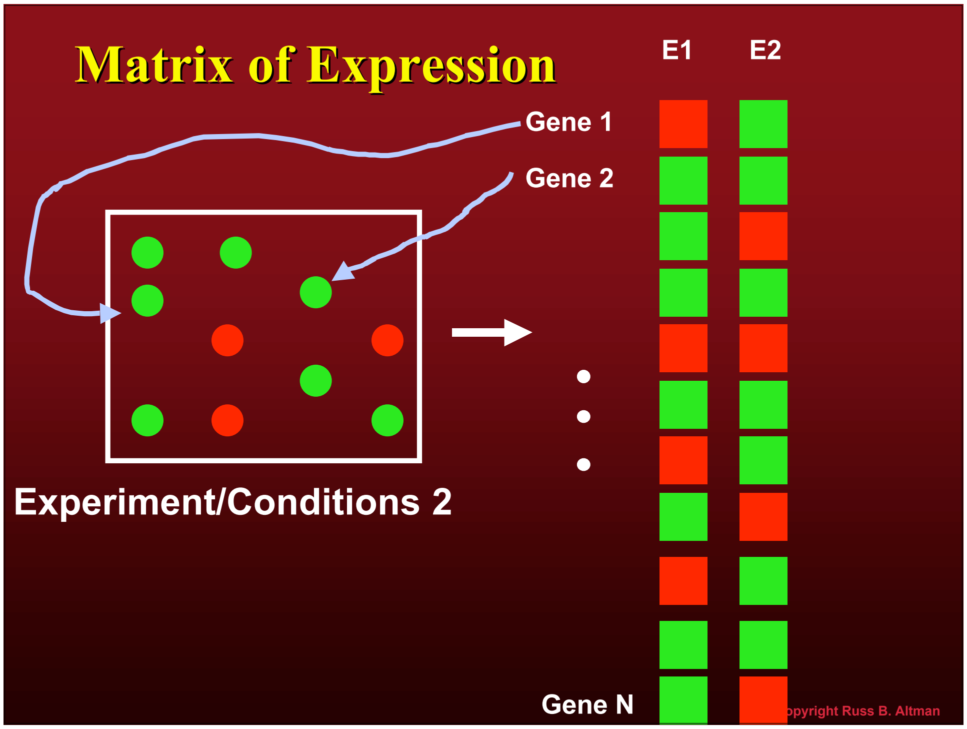


Matrix of Expression

Experiment/Conditions 1

	E1
Gene 1	🟥
Gene 2	🟩
	🟩
	🟩
	🟥
	🟩
	🟥
	🟩
	🟥
	🟩
Gene N	🟩

Copyright Russ B. Altman

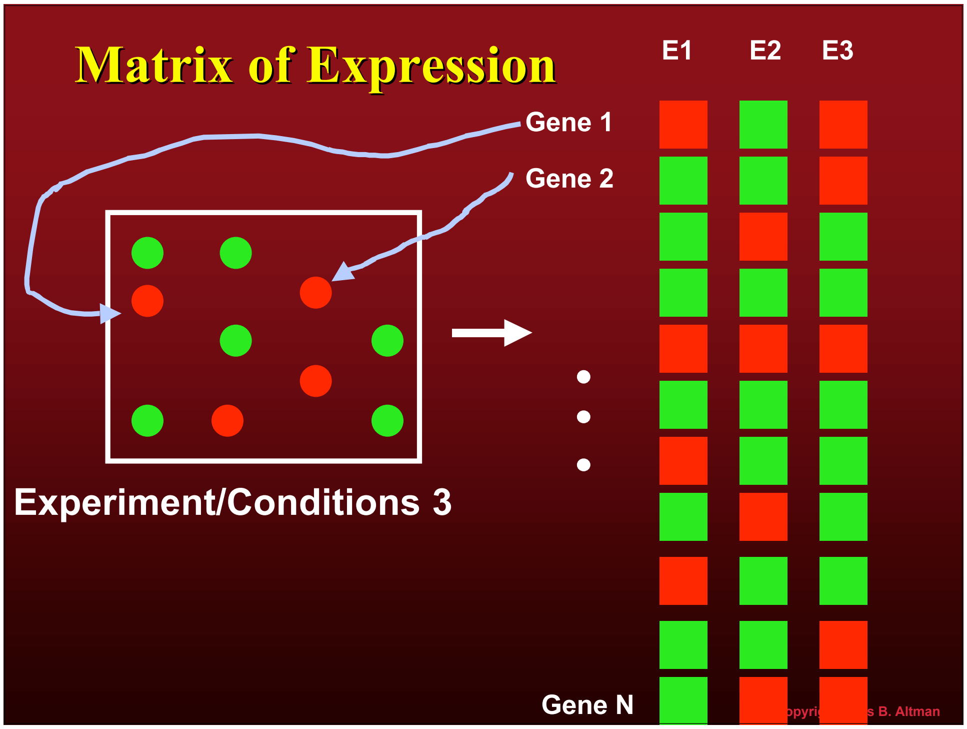


Matrix of Expression

Experiment/Conditions 2

	E1	E2
Gene 1	🟥	🟩
Gene 2	🟩	🟩
	🟩	🟥
	🟩	🟩
	🟥	🟥
	🟩	🟩
	🟥	🟩
	🟩	🟥
	🟥	🟩
	🟩	🟩
Gene N	🟩	🟥

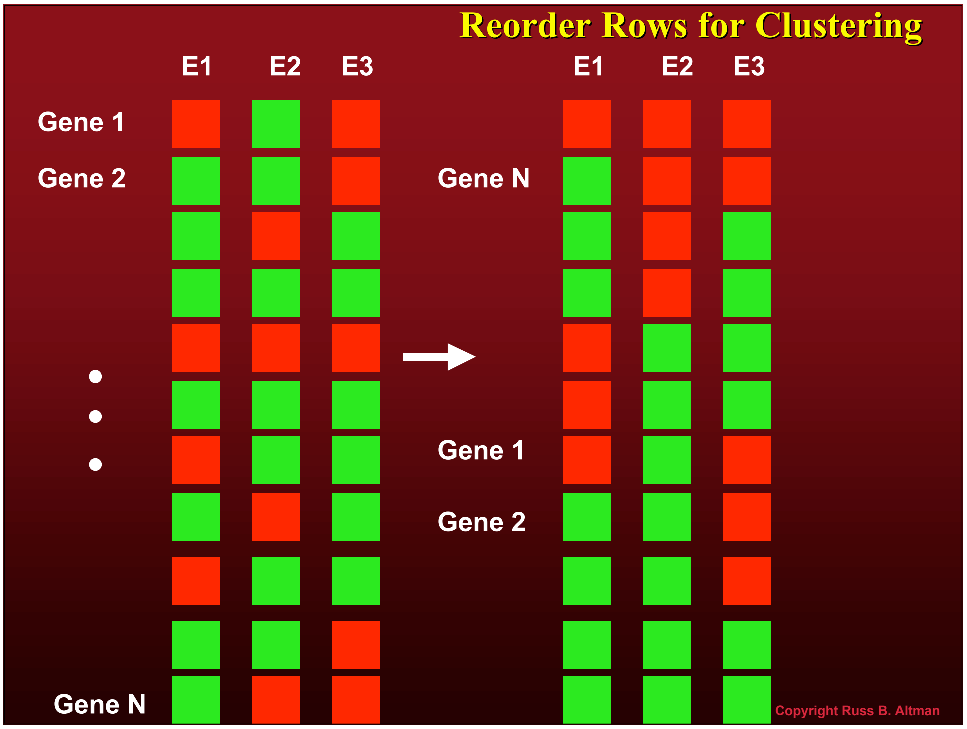
opyright Russ B. Altman



Matrix of Expression

Experiment/Conditions 3

	E1	E2	E3
Gene 1	🟥	🟩	🟥
Gene 2	🟩	🟩	🟥
•	🟩	🟥	🟩
•	🟩	🟩	🟩
•	🟥	🟥	🟥
	🟩	🟩	🟩
	🟥	🟩	🟩
	🟩	🟥	🟩
	🟥	🟩	🟩
	🟩	🟩	🟥
Gene N	🟩	🟥	🟥

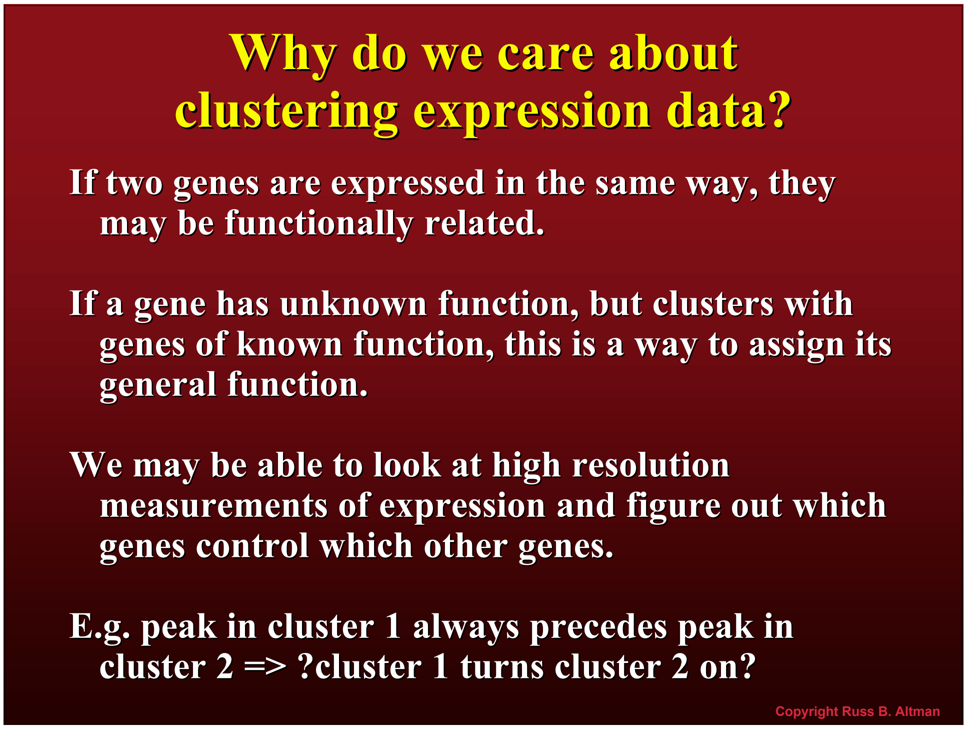


Reorder Rows for Clustering

	E1	E2	E3
Gene 1	🟥	🟩	🟥
Gene 2	🟩	🟩	🟥
	🟩	🟥	🟩
	🟩	🟩	🟩
	🟥	🟥	🟥
•	🟩	🟩	🟩
•	🟥	🟩	🟩
•	🟩	🟥	🟩
	🟥	🟩	🟩
	🟩	🟩	🟥
Gene N	🟩	🟥	🟥

→

	E1	E2	E3
	🟥	🟥	🟥
Gene N	🟩	🟥	🟥
	🟩	🟥	🟩
	🟩	🟥	🟩
	🟥	🟩	🟩
	🟥	🟩	🟩
Gene 1	🟥	🟩	🟥
Gene 2	🟩	🟩	🟥
	🟩	🟩	🟥
	🟩	🟩	🟩
	🟩	🟩	🟩



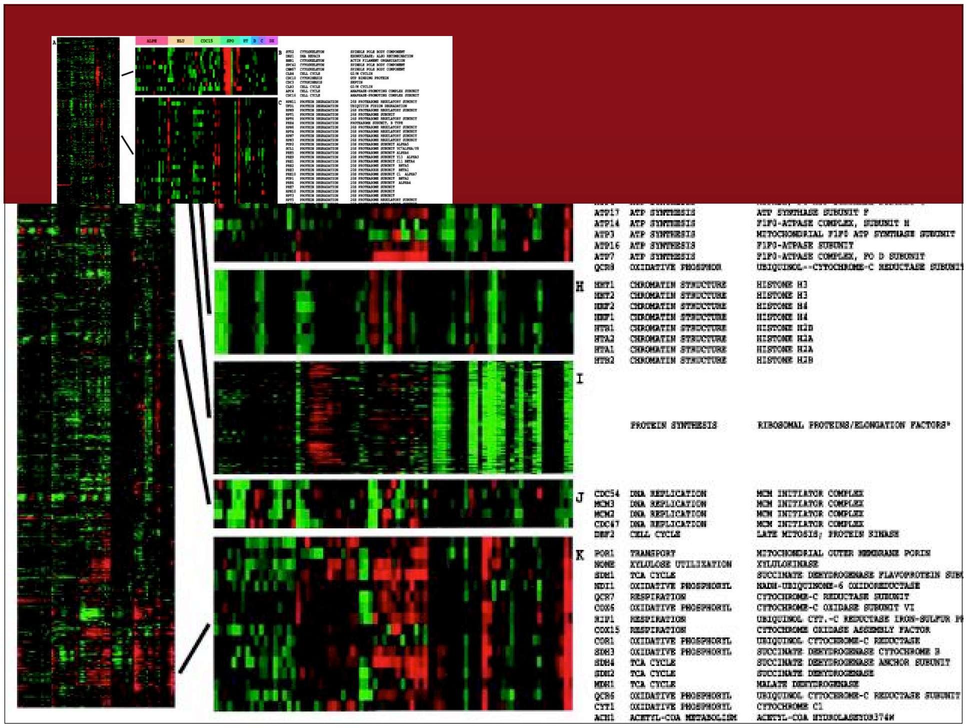
Why do we care about clustering expression data?

If two genes are expressed in the same way, they may be functionally related.

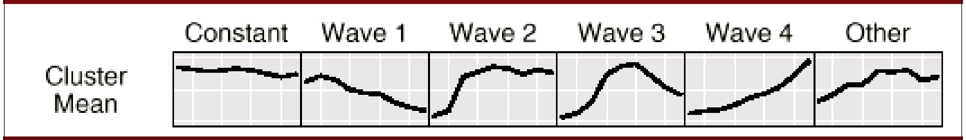
If a gene has unknown function, but clusters with genes of known function, this is a way to assign its general function.

We may be able to look at high resolution measurements of expression and figure out which genes control which other genes.

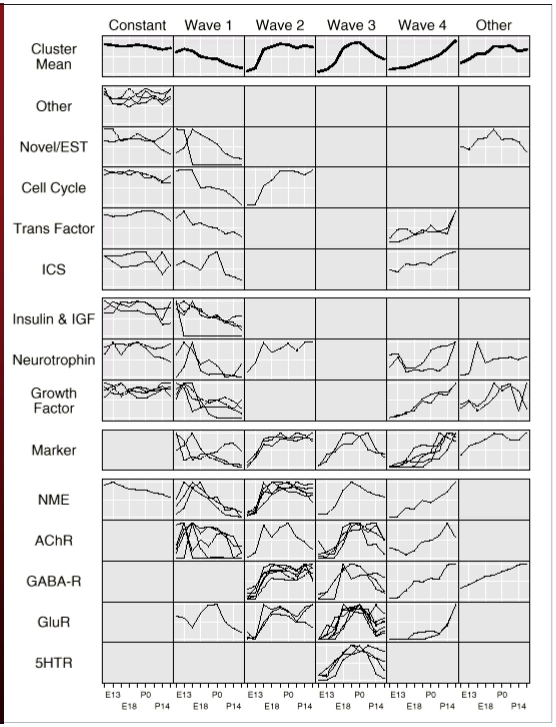
E.g. peak in cluster 1 always precedes peak in cluster 2 => ?cluster 1 turns cluster 2 on?

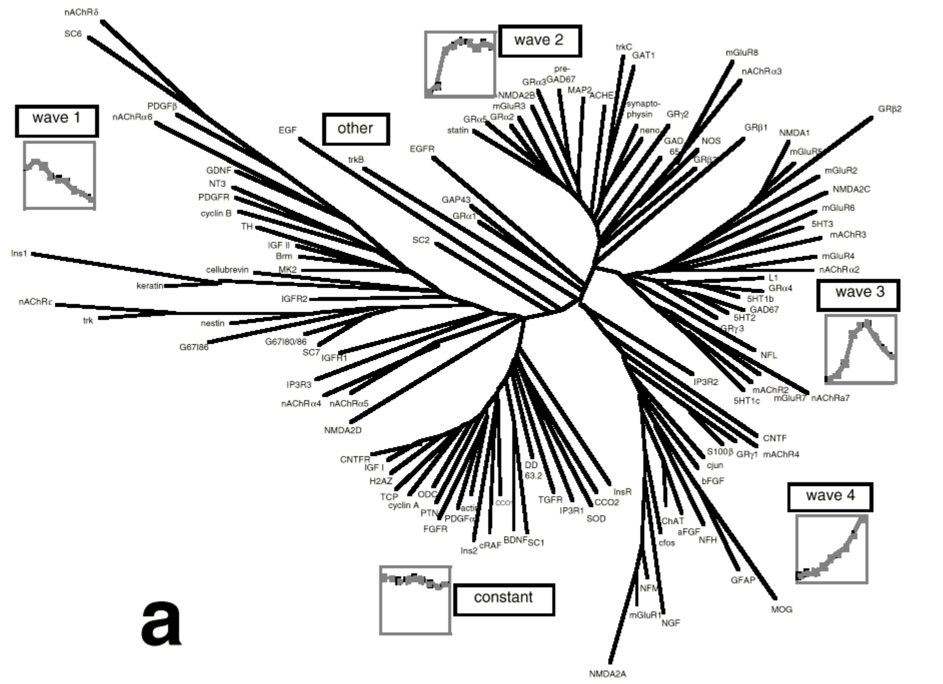


Average of clustered wave forms



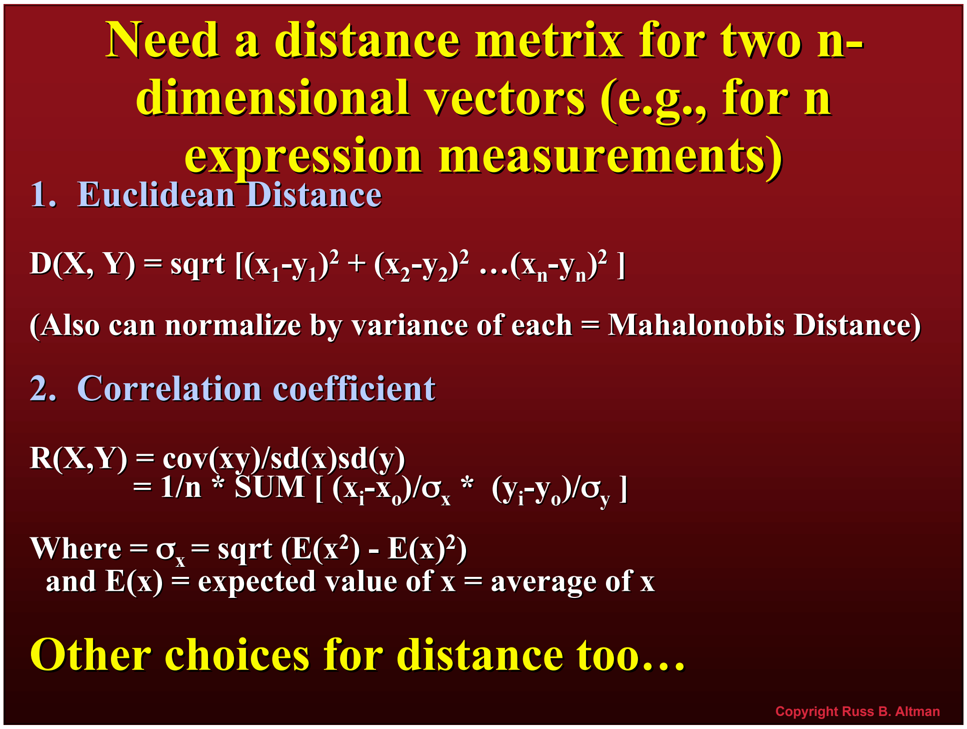
Typical "wave forms" observed (note: not lots of bumps)





Methods for Clustering

- K-means
- Hierarchical Clustering
- Self Organizing Maps
- Trillions of others.



Need a distance metrix for two n-dimensional vectors (e.g., for n expression measurements)

1. Euclidean Distance

$D(X, Y) = \text{sqrt}\ [(x_1-y_1)^2 + (x_2-y_2)^2 \ldots(x_n-y_n)^2\ ]$

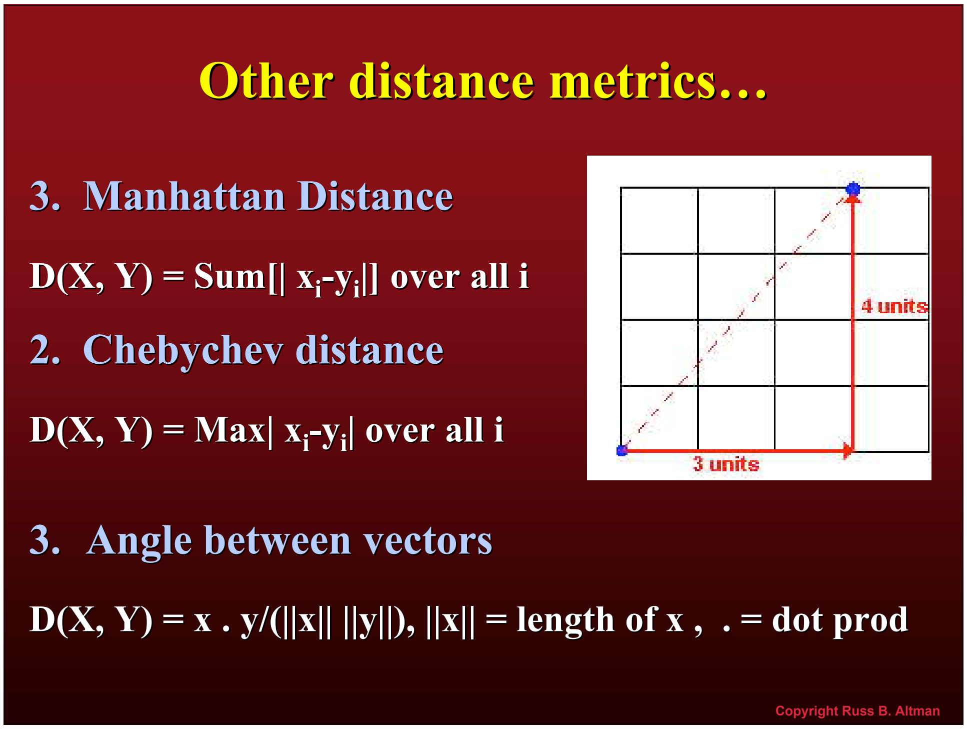
(Also can normalize by variance of each = Mahalonobis Distance)

2. Correlation coefficient

$R(X,Y) = \text{cov}(xy)/\text{sd}(x)\text{sd}(y)$
$\quad = 1/n * \text{SUM}\ [\ (x_i-x_o)/\sigma_x * \ (y_i-y_o)/\sigma_y\ ]$

Where = $\sigma_x = \text{sqrt}\ (E(x^2) - E(x)^2)$
and $E(x)$ = expected value of x = average of x

Other choices for distance too…


4 units
3 units

Other distance metrics…

3. Manhattan Distance

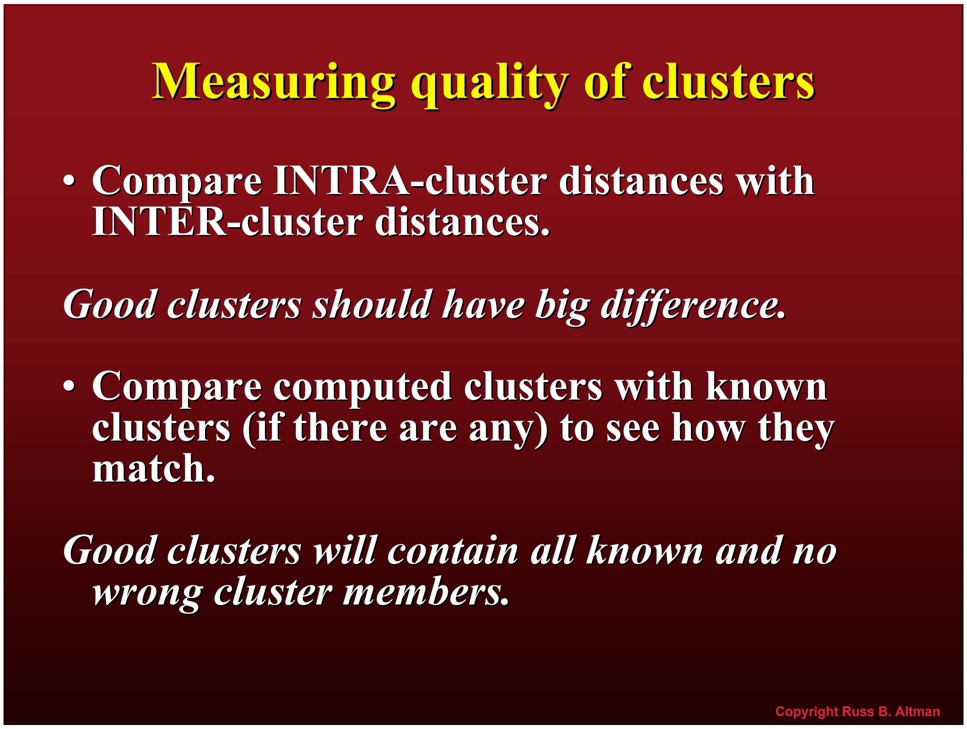
$D(X, Y) = \text{Sum}[|\ x_i-y_i|]$ over all i

2. Chebychev distance

$D(X, Y) = \text{Max}|\ x_i-y_i|$ over all i

3. Angle between vectors

$D(X, Y) = x\ .\ y/(||x||\ ||y||)$, $||x||$ = length of x , . = dot prod



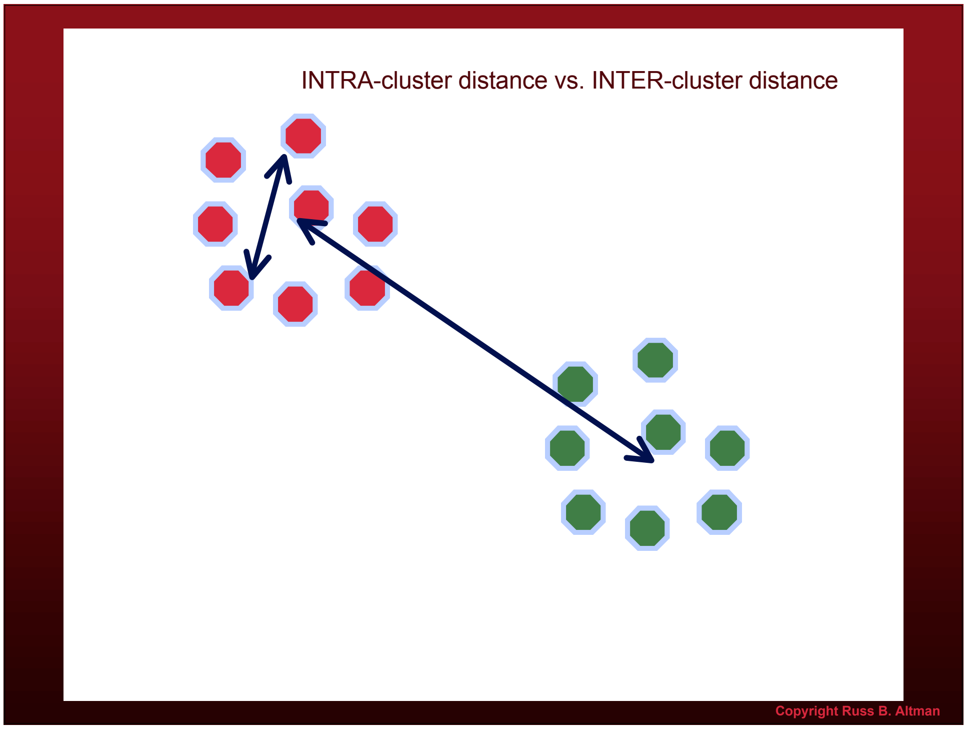
Measuring quality of clusters

- Compare INTRA-cluster distances with INTER-cluster distances.

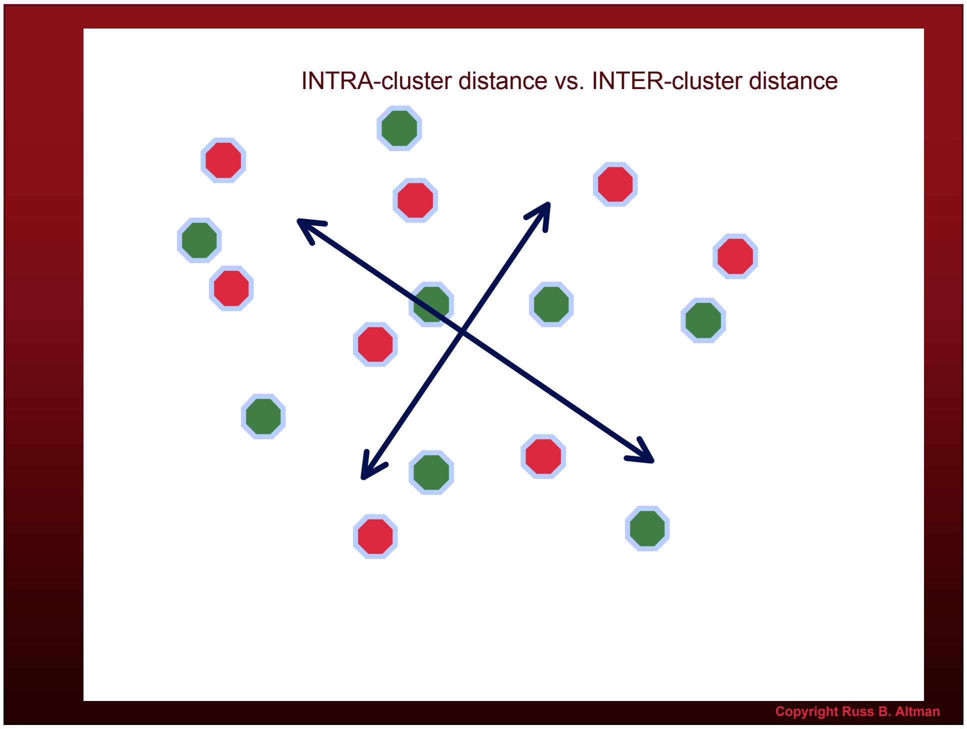
Good clusters should have big difference.

- Compare computed clusters with known clusters (if there are any) to see how they match.

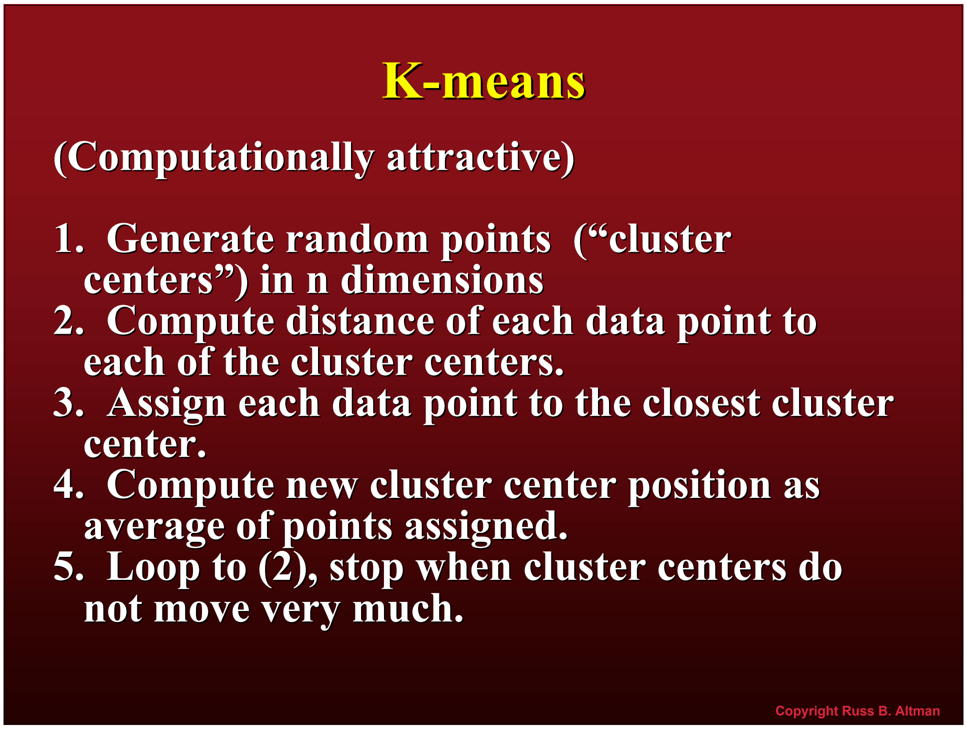
Good clusters will contain all known and no wrong cluster members.



INTRA-cluster distance vs. INTER-cluster distance



INTRA-cluster distance vs. INTER-cluster distance



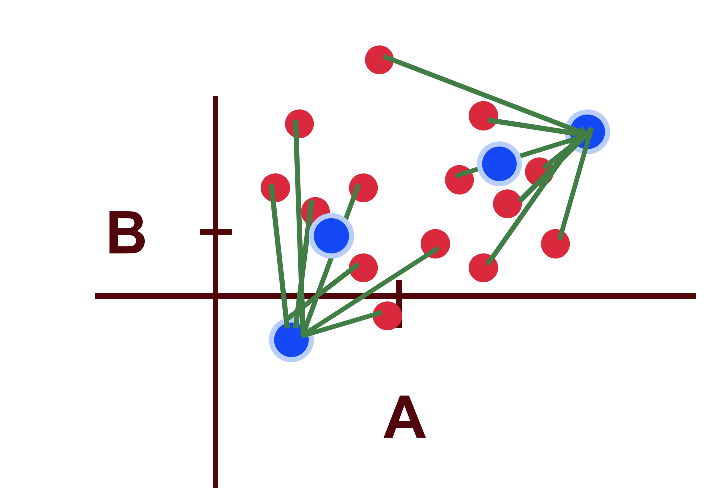
K-means

(Computationally attractive)

1. Generate random points ("cluster centers") in n dimensions
2. Compute distance of each data point to each of the cluster centers.
3. Assign each data point to the closest cluster center.
4. Compute new cluster center position as average of points assigned.
5. Loop to (2), stop when cluster centers do not move very much.

Graphical Representation

Two features f1 (x-coordinate) and f2 (y-coordinate)

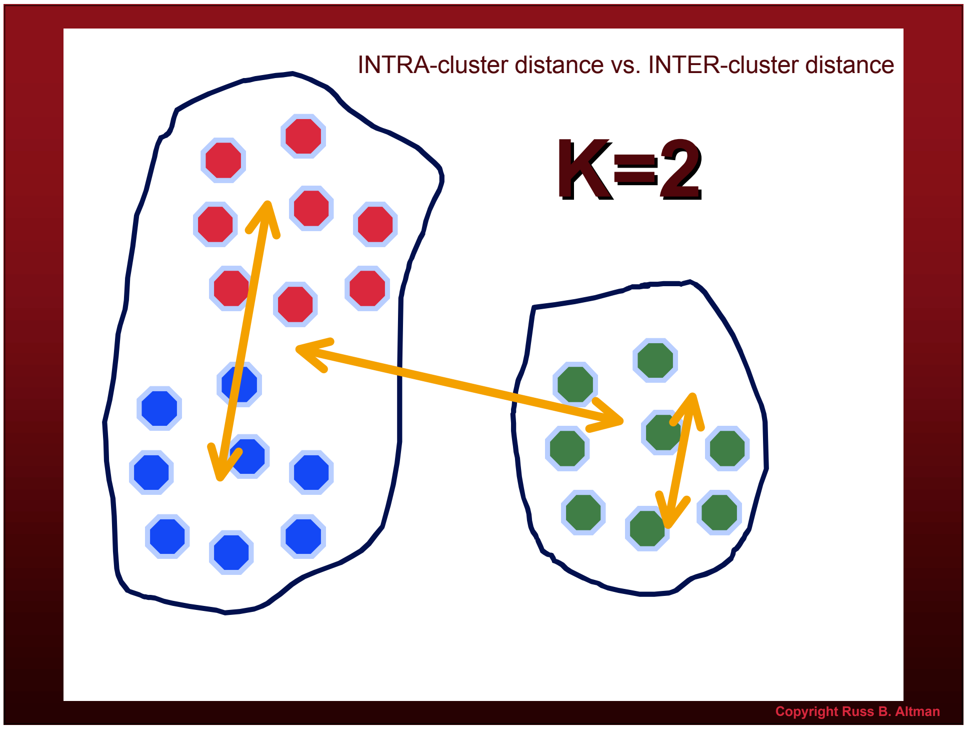


Copyright Russ B. Altman

K-means issues

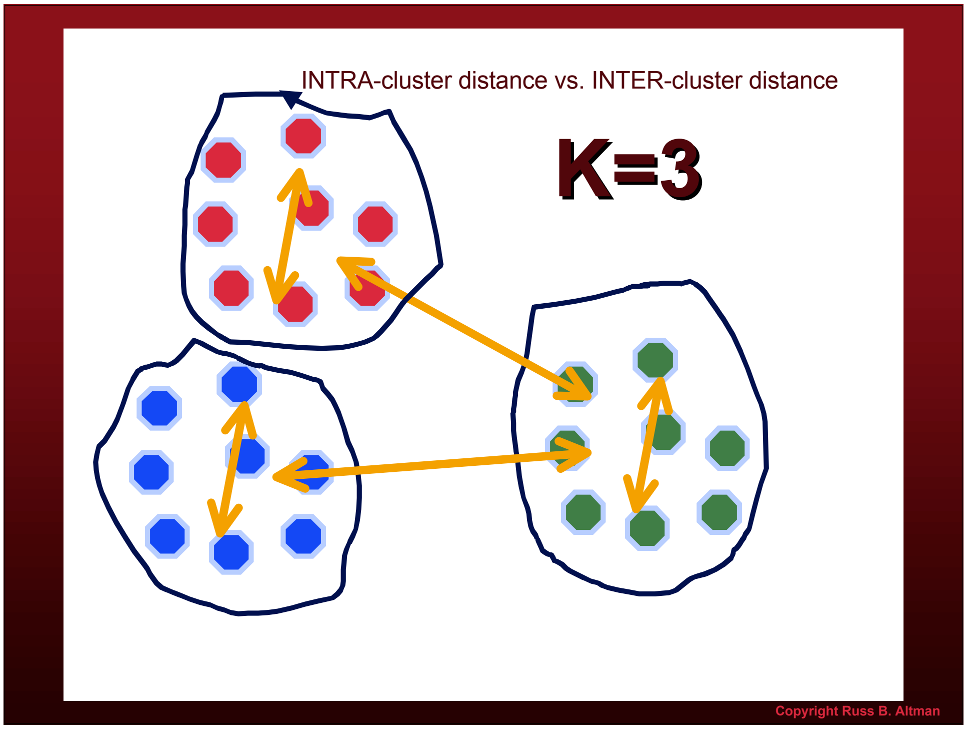
- Fast, O(N)
- Hard to know what K to choose
  - Try a bunch, and measure quality
- Hard to know where to seed the clusters
- Results can change drastically with different seeds.

Copyright Russ B. Altman



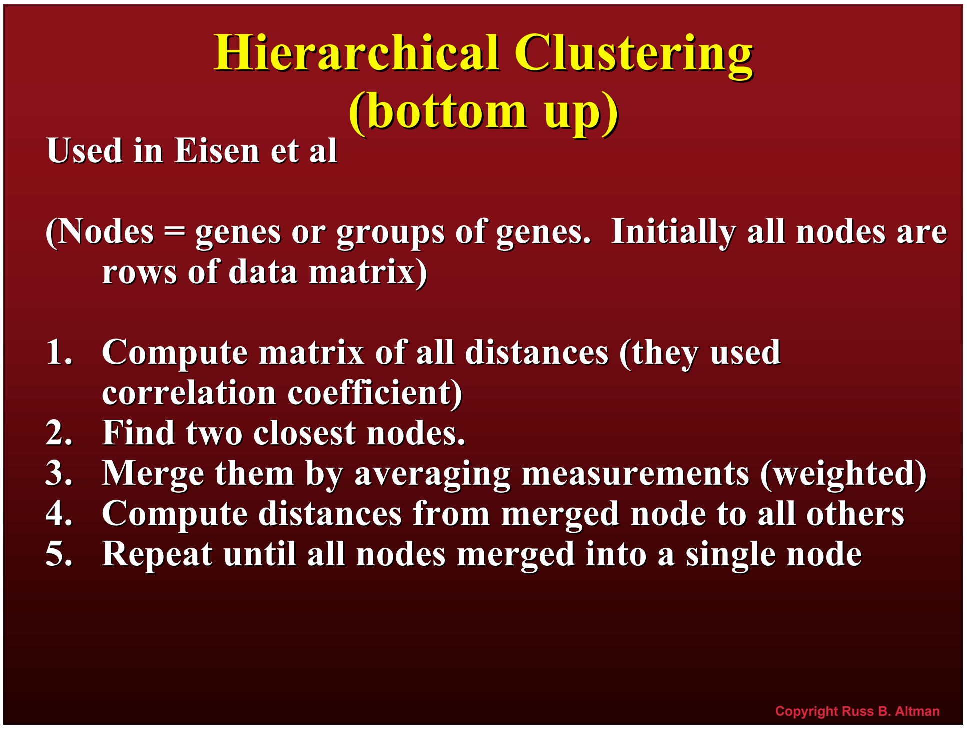
INTRA-cluster distance vs. INTER-cluster distance

K=2



INTRA-cluster distance vs. INTER-cluster distance

K=3

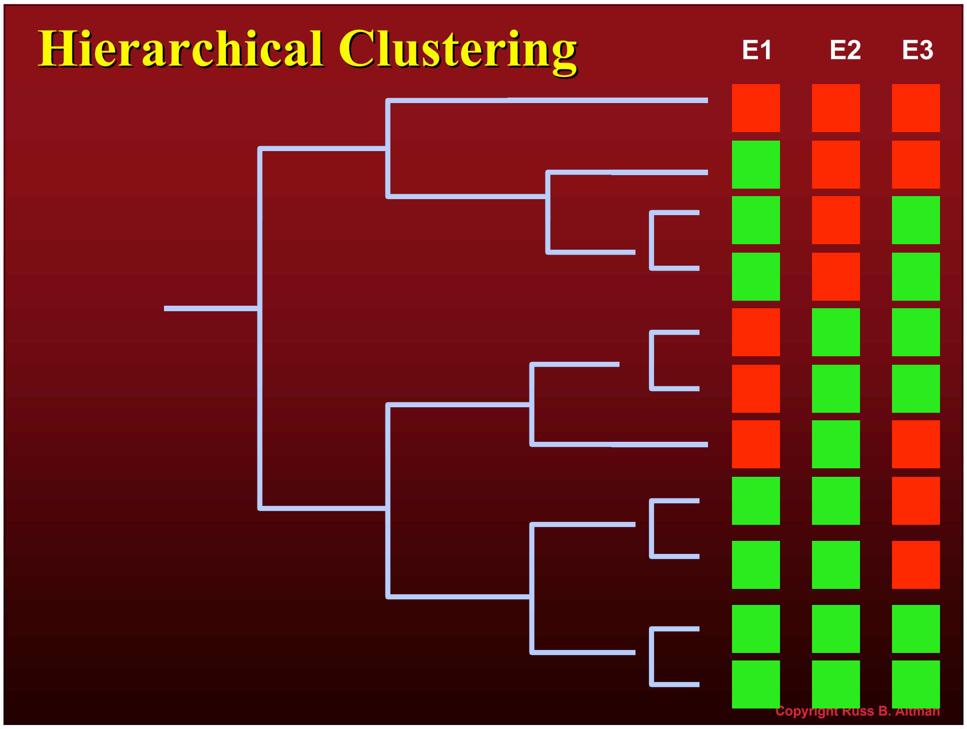


Hierarchical Clustering (bottom up)

Used in Eisen et al

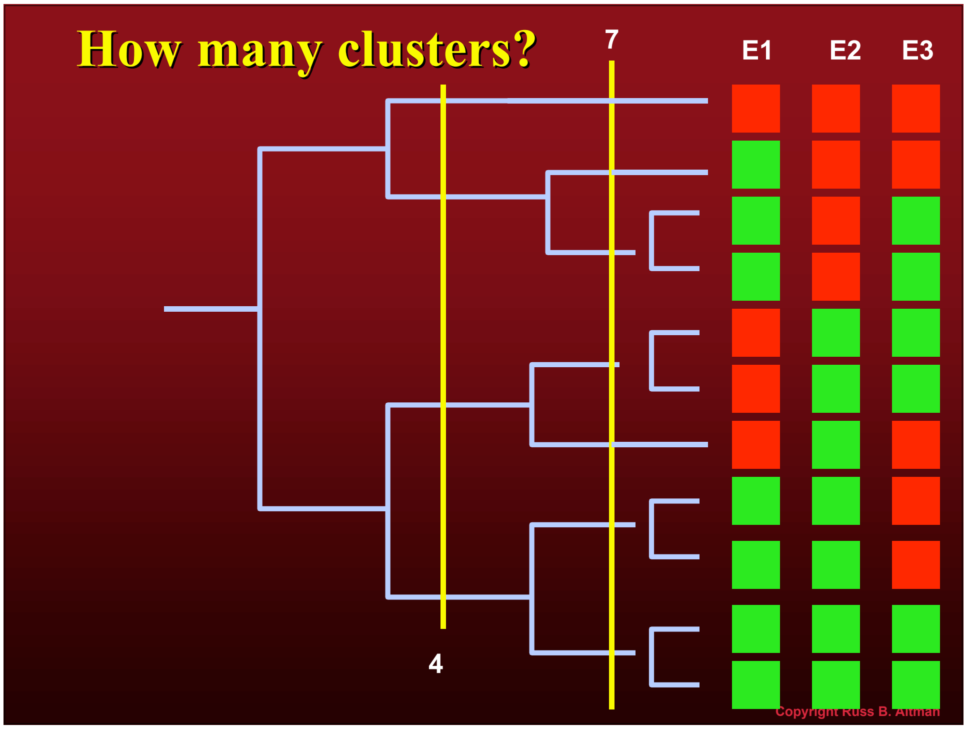
(Nodes = genes or groups of genes. Initially all nodes are rows of data matrix)

1. Compute matrix of all distances (they used correlation coefficient)
2. Find two closest nodes.
3. Merge them by averaging measurements (weighted)
4. Compute distances from merged node to all others
5. Repeat until all nodes merged into a single node



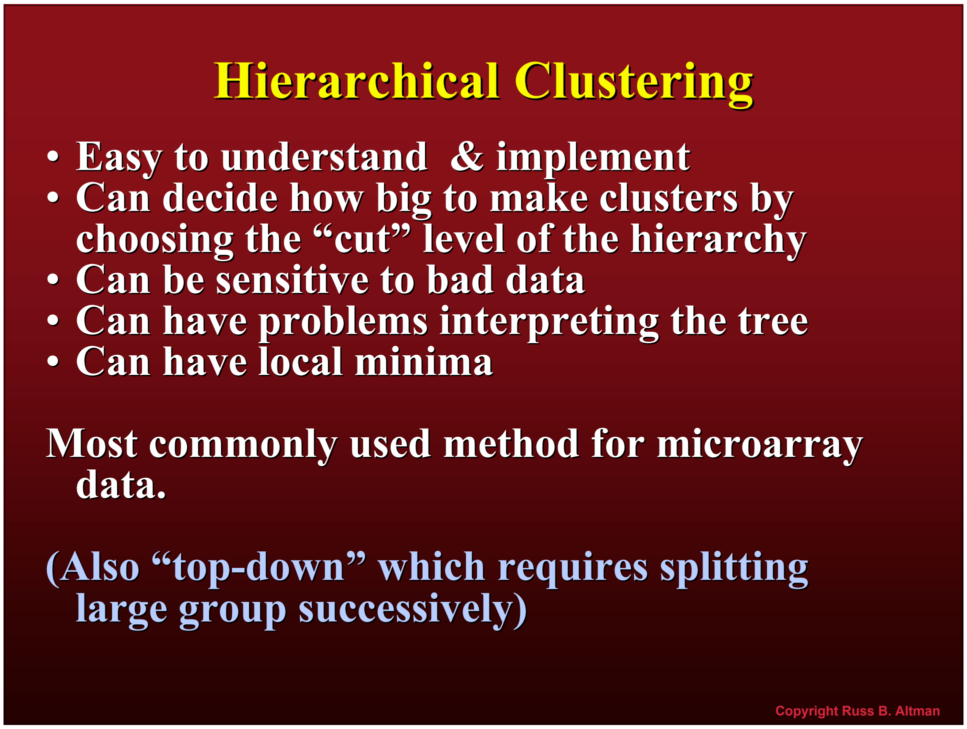
Hierarchical Clustering

E1	E2	E3
🟥	🟥	🟥
🟩	🟥	🟥
🟩	🟥	🟩
🟩	🟥	🟩
🟥	🟩	🟩
🟥	🟩	🟩
🟥	🟩	🟥
🟩	🟩	🟥
🟩	🟩	🟥
🟩	🟩	🟩
🟩	🟩	🟩



How many clusters?

E1	E2	E3
🟥	🟥	🟥
🟩	🟥	🟥
🟩	🟥	🟩
🟩	🟥	🟩
🟥	🟩	🟩
🟥	🟩	🟩
🟥	🟩	🟥
🟩	🟩	🟥
🟩	🟩	🟥
🟩	🟩	🟩
🟩	🟩	🟩



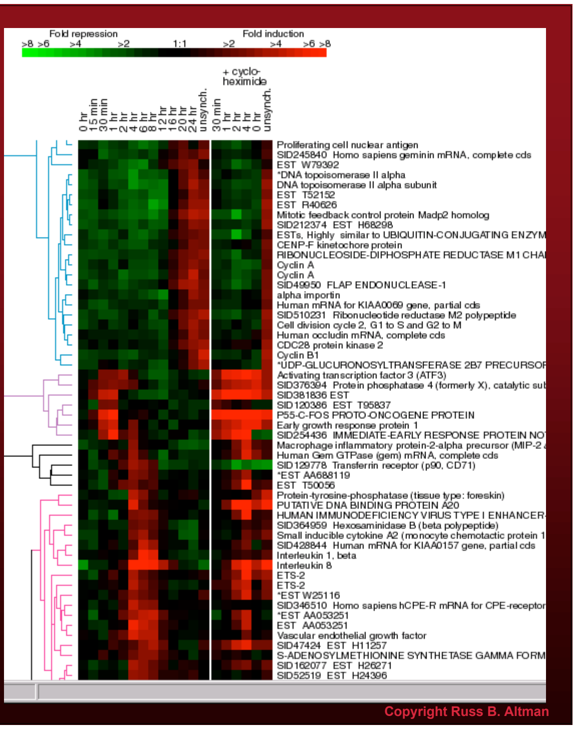
Hierarchical Clustering

- Easy to understand & implement
- Can decide how big to make clusters by choosing the “cut” level of the hierarchy
- Can be sensitive to bad data
- Can have problems interpreting the tree
- Can have local minima

Most commonly used method for microarray data.

(Also “top-down” which requires splitting large group successively)

Can build trees from cluster analysis, groups genes by common patterns of expression.



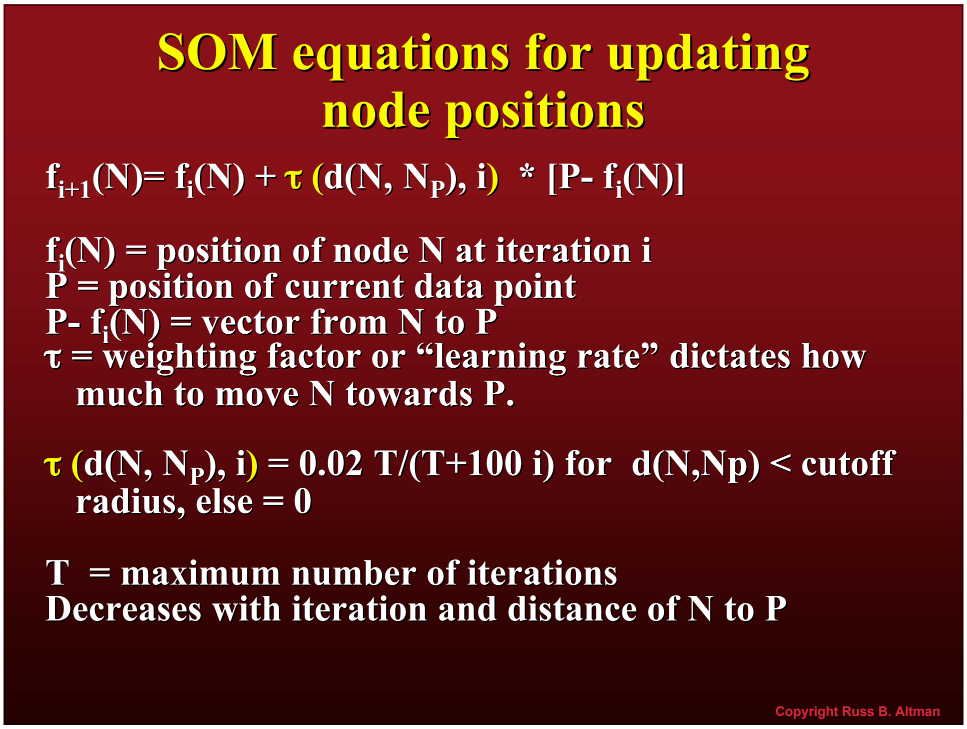
Self Organizing Maps

Used by Tamayo et al
(use same idea of nodes)

1. Generate a simple (usually) 2D grid of nodes (x,y)
2. Map the nodes into n-dim expression vectors (initially randomly)

(e.g. (x,y) -> [0 0 0 x 0 0 0 y 0 0 0 0 0])

3. For each data point, P, change all *node positions* so that they move towards P. Closer nodes move more than far nodes.
4. Iterate for a maximum number of iterations, and then assess position of all nodes.



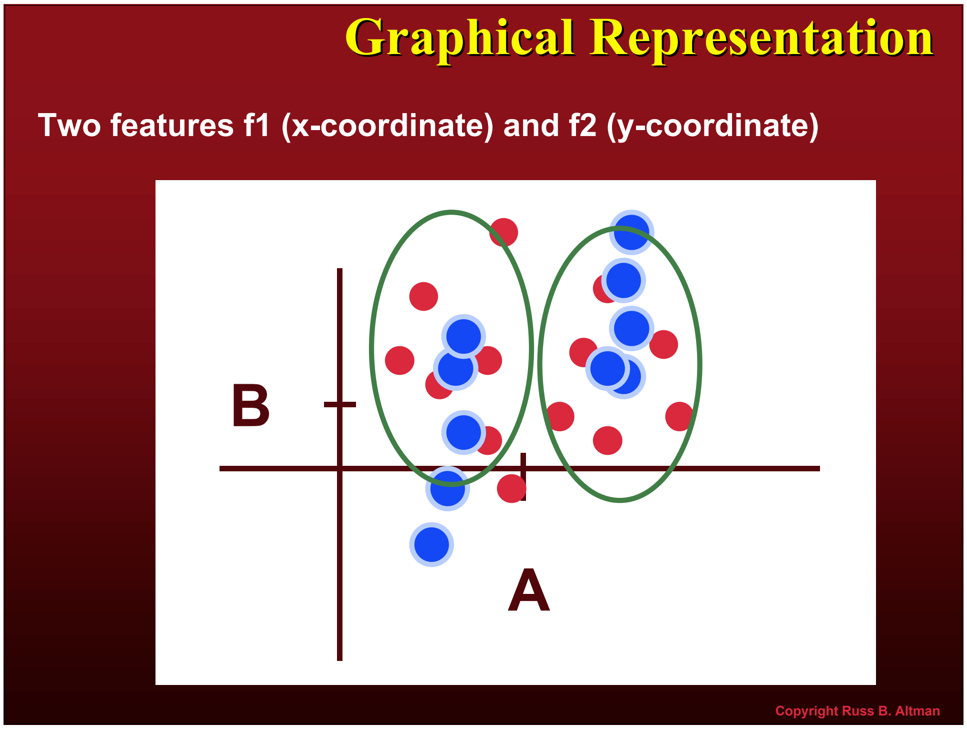
SOM equations for updating node positions

$f_{i+1}(N)= f_i(N) + \tau\ (d(N, N_P), i)\ \ * [P- f_i(N)]$

$f_i(N)$ = position of node N at iteration i
P = position of current data point
$P- f_i(N)$ = vector from N to P
τ = weighting factor or "learning rate" dictates how much to move N towards P.

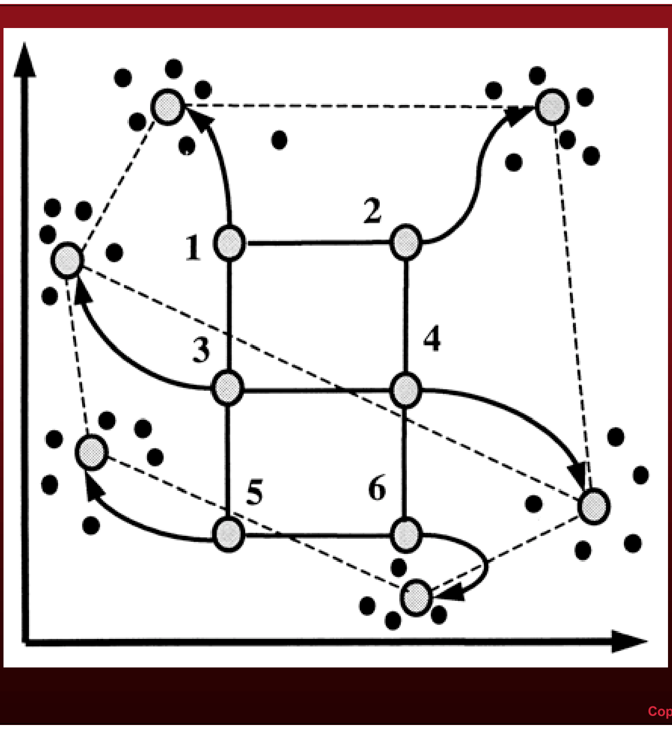
$\tau\ (d(N, N_P), i) = 0.02\ T/(T+100\ i)$ for $d(N,Np) <$ cutoff radius, else = 0

T = maximum number of iterations
Decreases with iteration and distance of N to P



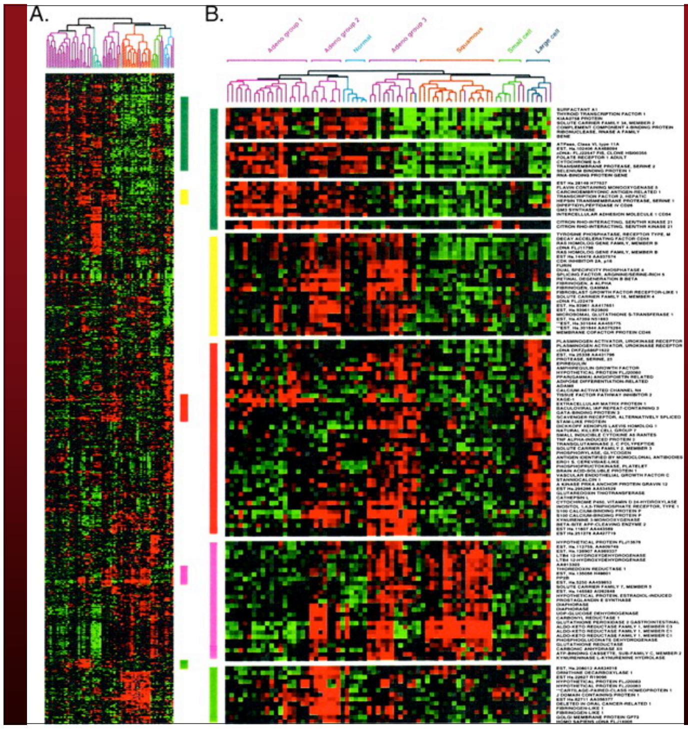
Graphical Representation

Two features f1 (x-coordinate) and f2 (y-coordinate)

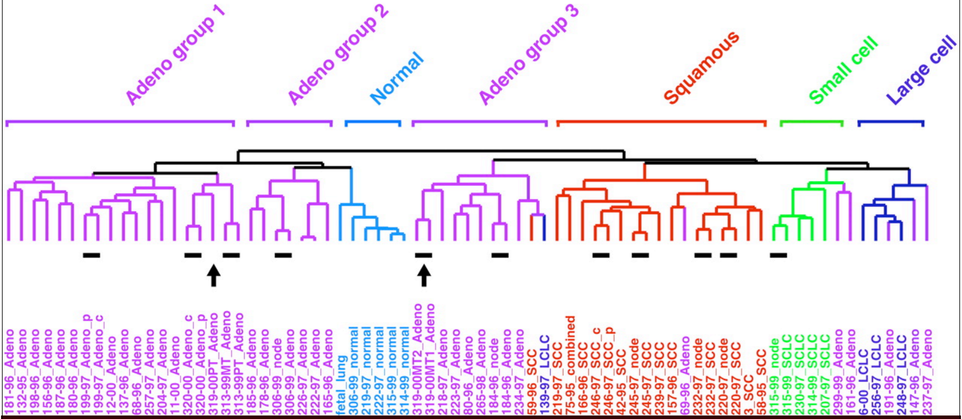


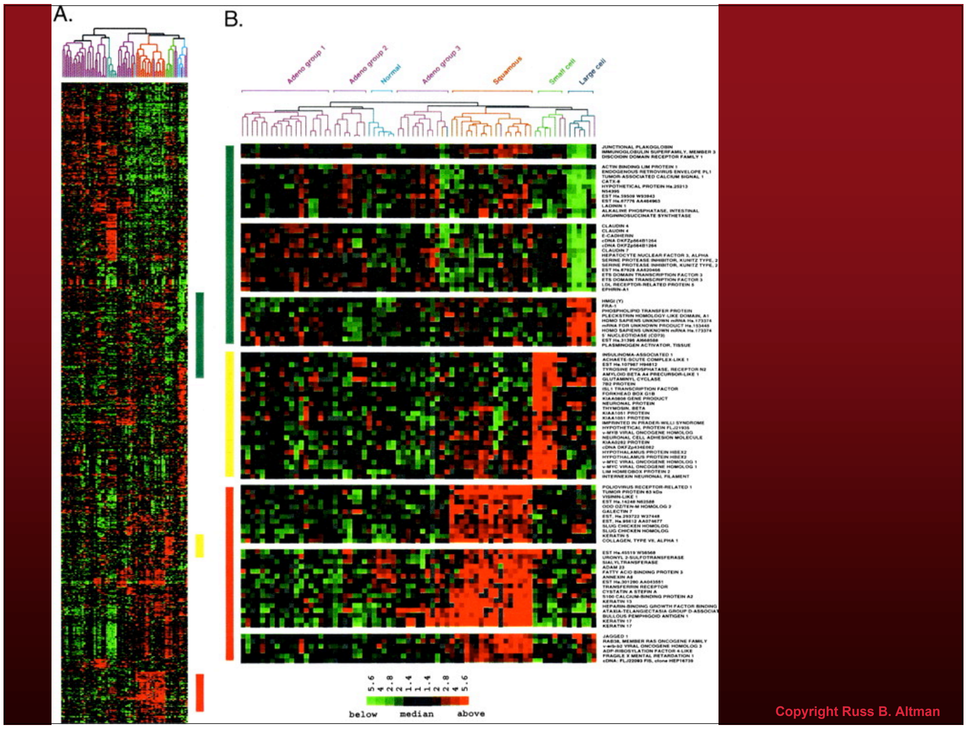
SOMs

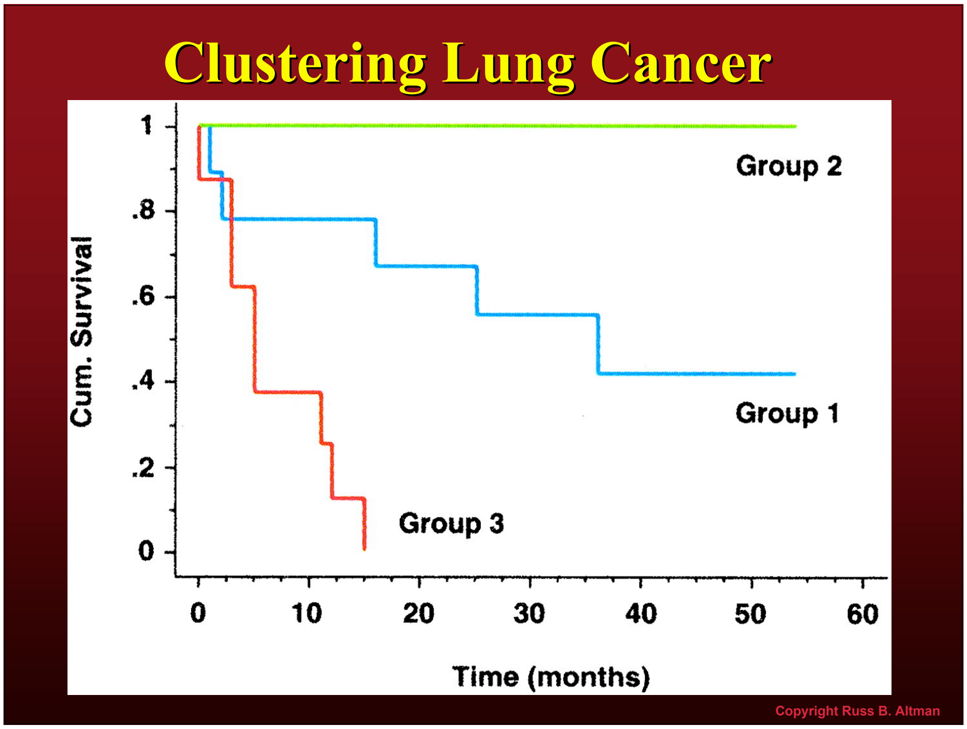
- Impose a partial structure on the cluster problem as a start
- Easy to implement
- Pretty fast
- Let the clusters move towards the data
- Easy to visualize results
- Can be sensitive to starting structure
- No guarantee of convergence to good clusters.

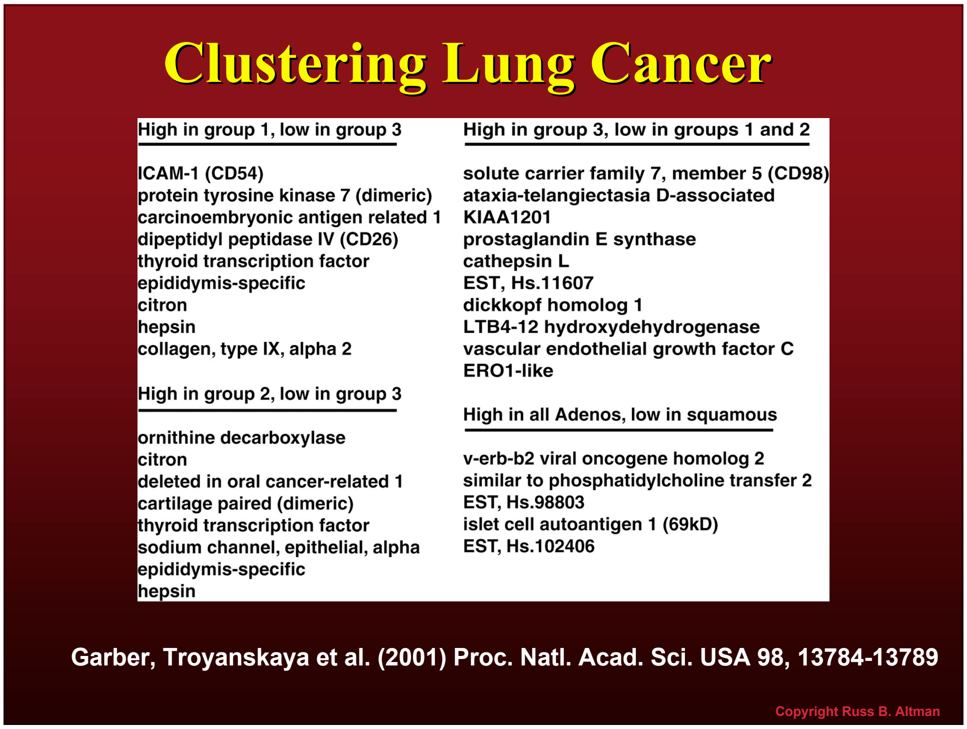


1. Similar lung cancers cluster together.
2. Cancers from same person cluster together.






Clustering Lung Cancer



Clustering Lung Cancer

High in group 1, low in group 3

ICAM-1 (CD54)
protein tyrosine kinase 7 (dimeric)
carcinoembryonic antigen related 1
dipeptidyl peptidase IV (CD26)
thyroid transcription factor
epididymis-specific
citron
hepsin
collagen, type IX, alpha 2

High in group 2, low in group 3

ornithine decarboxylase
citron
deleted in oral cancer-related 1
cartilage paired (dimeric)
thyroid transcription factor
sodium channel, epithelial, alpha
epididymis-specific
hepsin

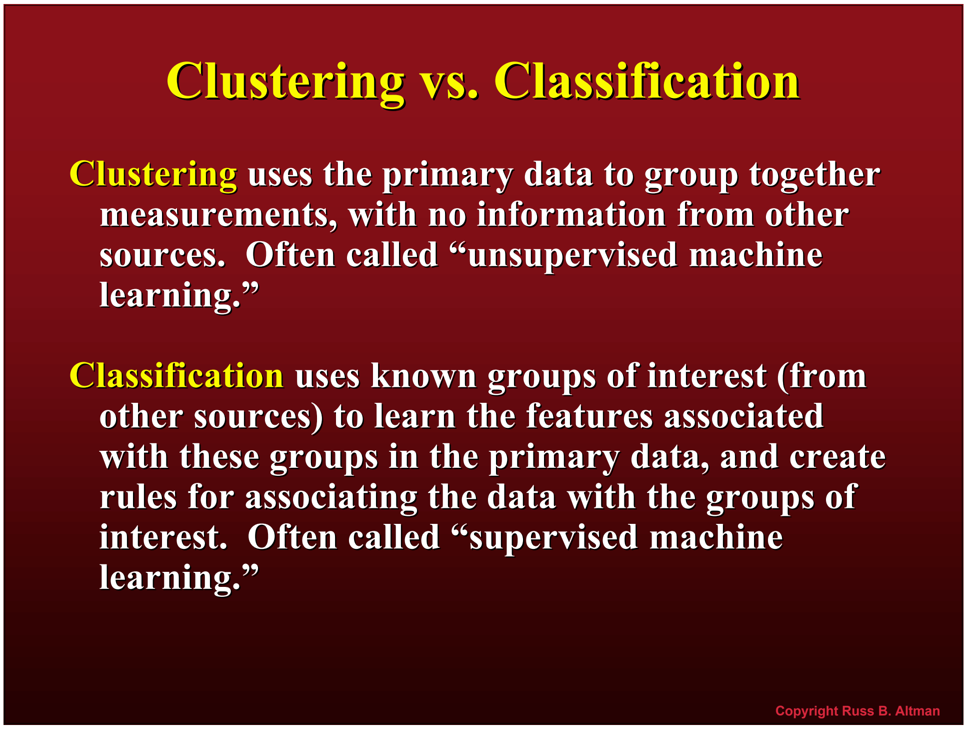
High in group 3, low in groups 1 and 2

solute carrier family 7, member 5 (CD98)
ataxia-telangiectasia D-associated
KIAA1201
prostaglandin E synthase
cathepsin L
EST, Hs.11607
dickkopf homolog 1
LTB4-12 hydroxydehydrogenase
vascular endothelial growth factor C
ERO1-like

High in all Adenos, low in squamous

v-erb-b2 viral oncogene homolog 2
similar to phosphatidylcholine transfer 2
EST, Hs.98803
islet cell autoantigen 1 (69kD)
EST, Hs.102406

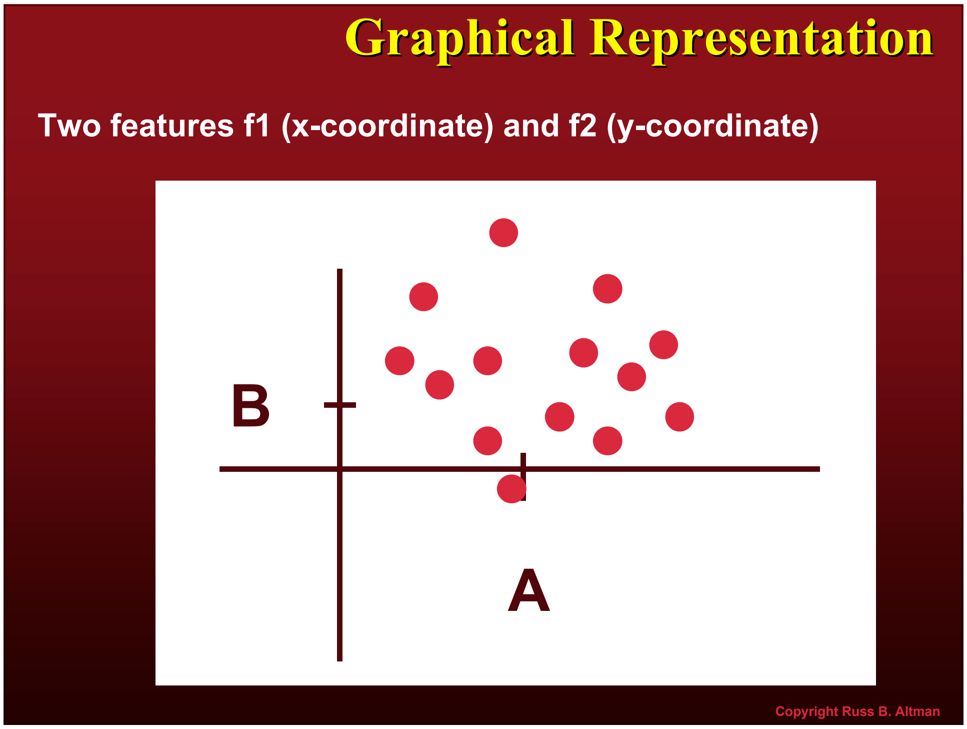
Garber, Troyanskaya et al. (2001) Proc. Natl. Acad. Sci. USA 98, 13784-13789



Clustering vs. Classification

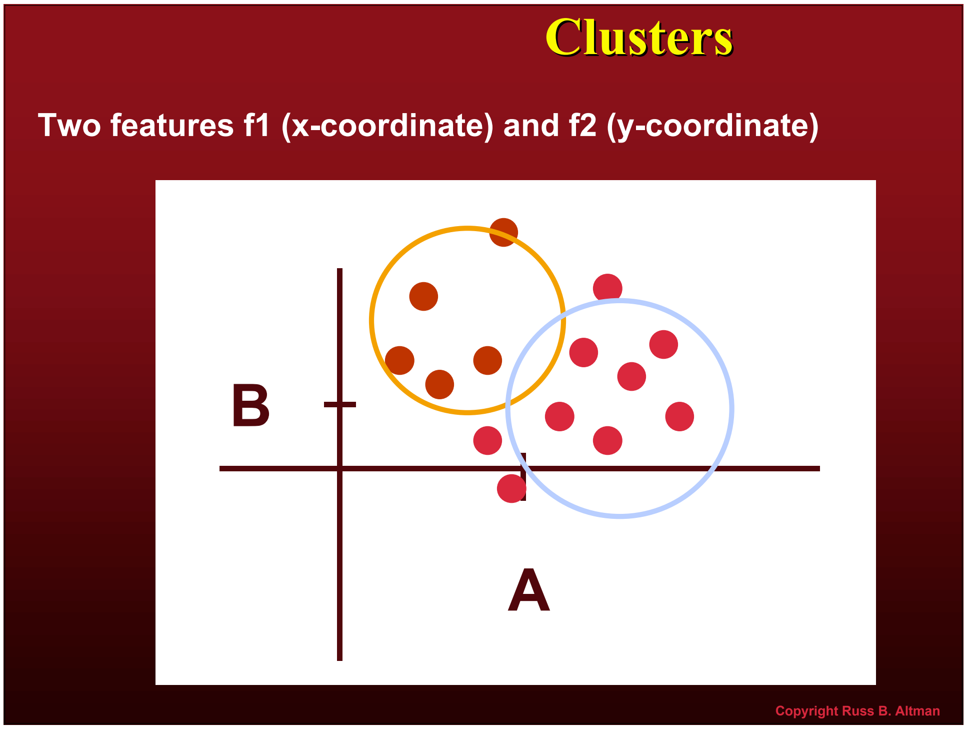
Clustering uses the primary data to group together measurements, with no information from other sources. Often called “unsupervised machine learning.”

Classification uses known groups of interest (from other sources) to learn the features associated with these groups in the primary data, and create rules for associating the data with the groups of interest. Often called “supervised machine learning.”



Graphical Representation

Two features f1 (x-coordinate) and f2 (y-coordinate)

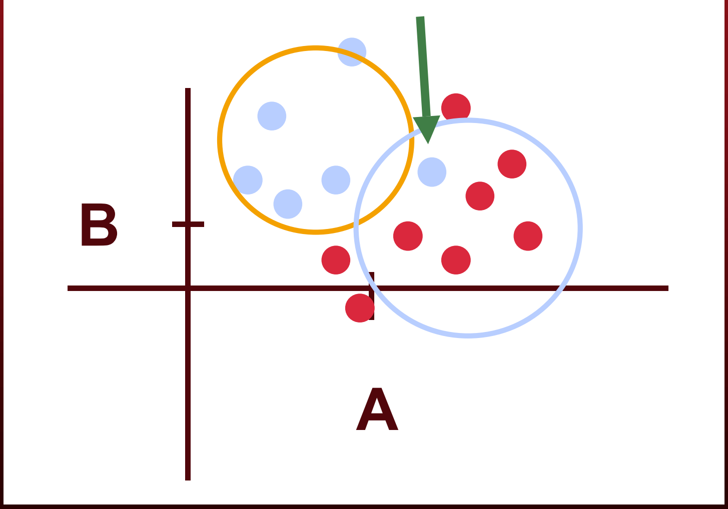


Clusters

Two features f1 (x-coordinate) and f2 (y-coordinate)

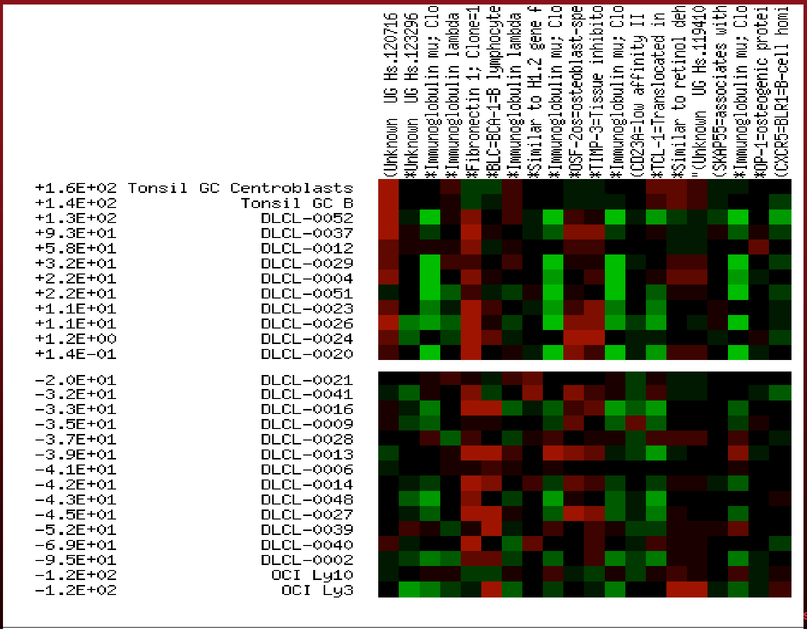
Apply external labels for classification

RED group and BLUE group now labeled

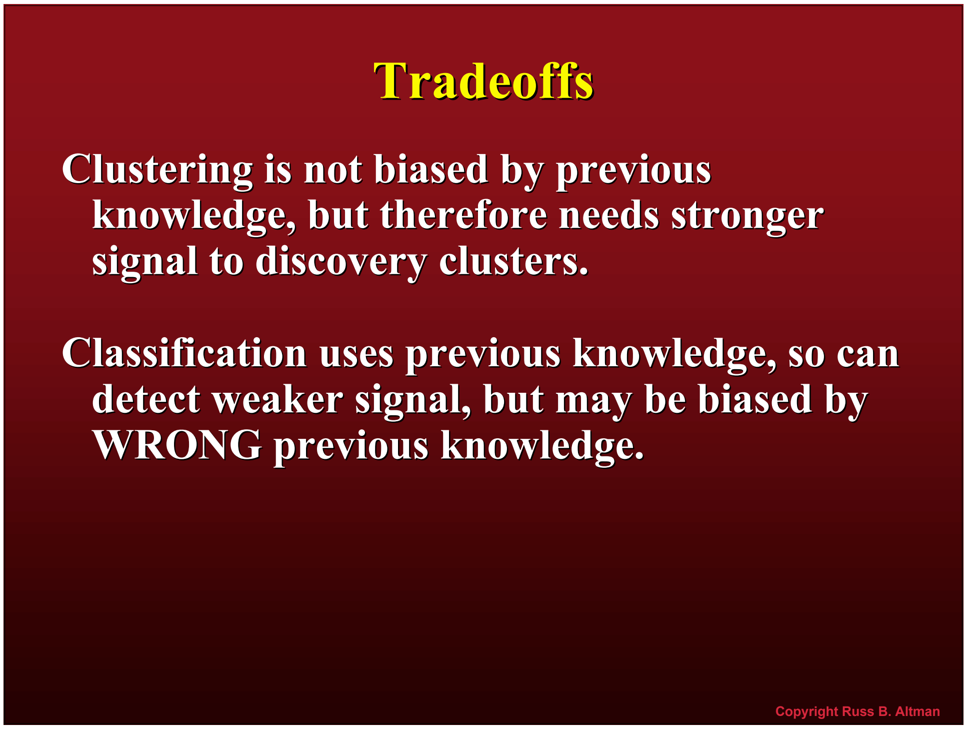


Copyright Russ B. Altman

Classifying Lymphomas



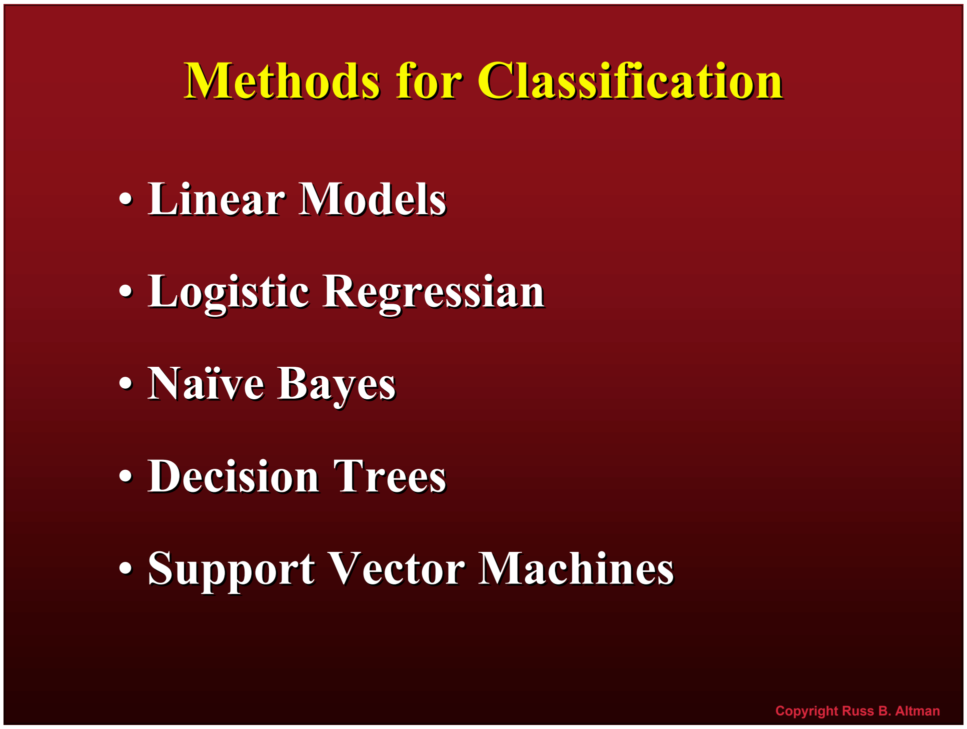
s B. Altman



Tradeoffs

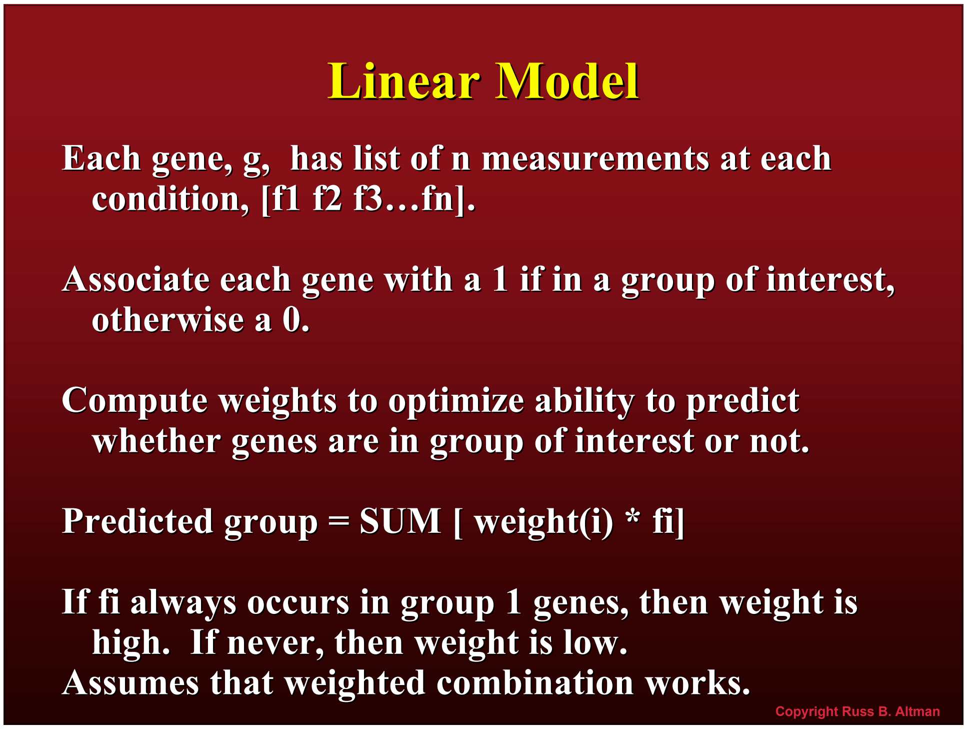
Clustering is not biased by previous knowledge, but therefore needs stronger signal to discovery clusters.

Classification uses previous knowledge, so can detect weaker signal, but may be biased by WRONG previous knowledge.



Methods for Classification

- Linear Models
- Logistic Regressian
- Naïve Bayes
- Decision Trees
- Support Vector Machines



Linear Model

Each gene, g, has list of n measurements at each condition, [f1 f2 f3…fn].

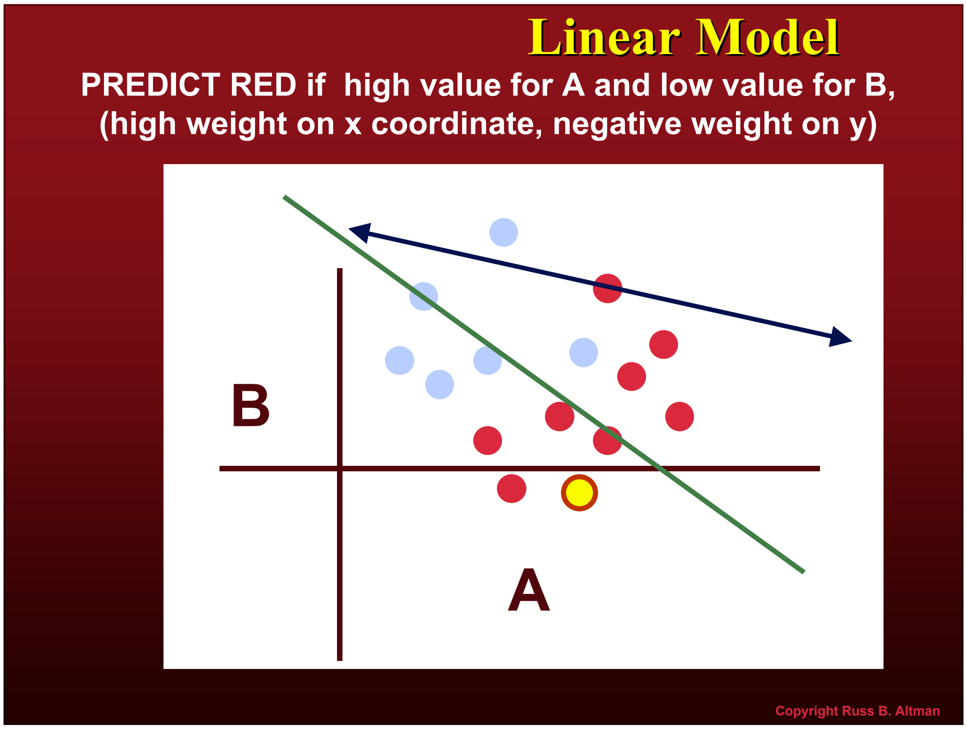
Associate each gene with a 1 if in a group of interest, otherwise a 0.

Compute weights to optimize ability to predict whether genes are in group of interest or not.

Predicted group = SUM [weight(i) * fi]

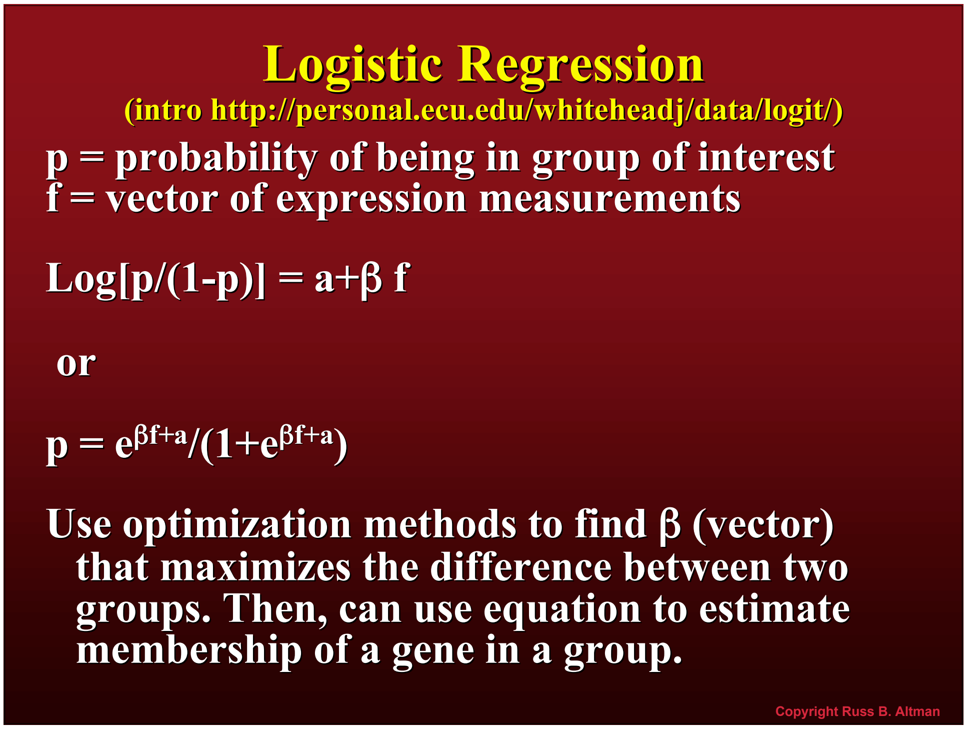
If fi always occurs in group 1 genes, then weight is high. If never, then weight is low.

Assumes that weighted combination works.



Linear Model

PREDICT RED if high value for A and low value for B, (high weight on x coordinate, negative weight on y)



Logistic Regression

(intro http://personal.ecu.edu/whiteheadj/data/logit/)

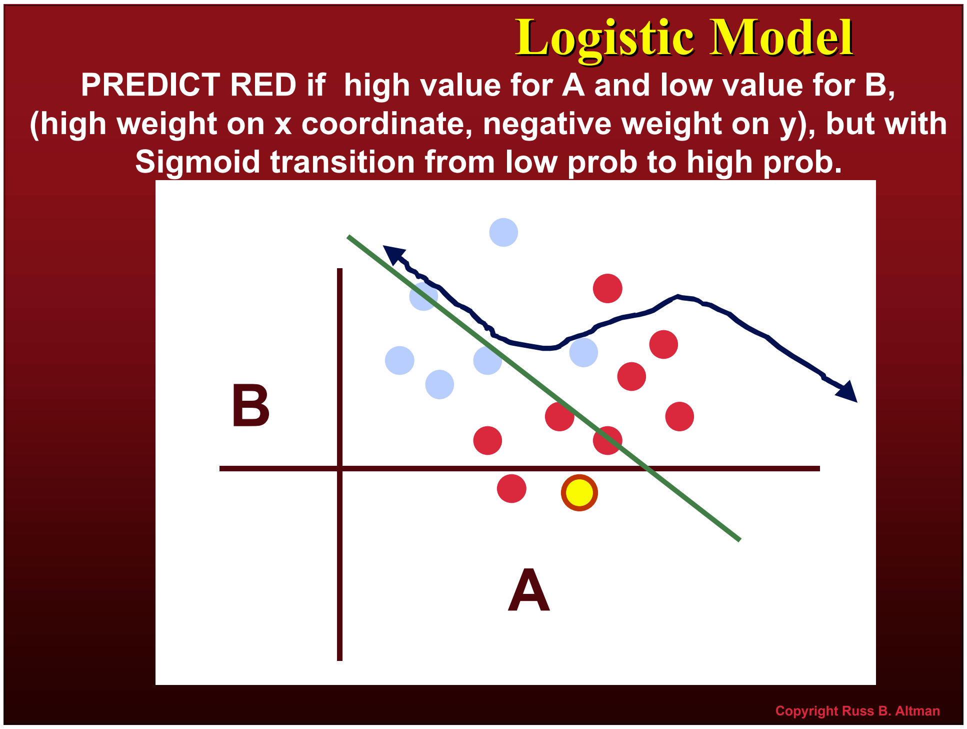
p = probability of being in group of interest
f = vector of expression measurements

$\mathrm{Log}[p/(1-p)] = a+\beta\, f$

or

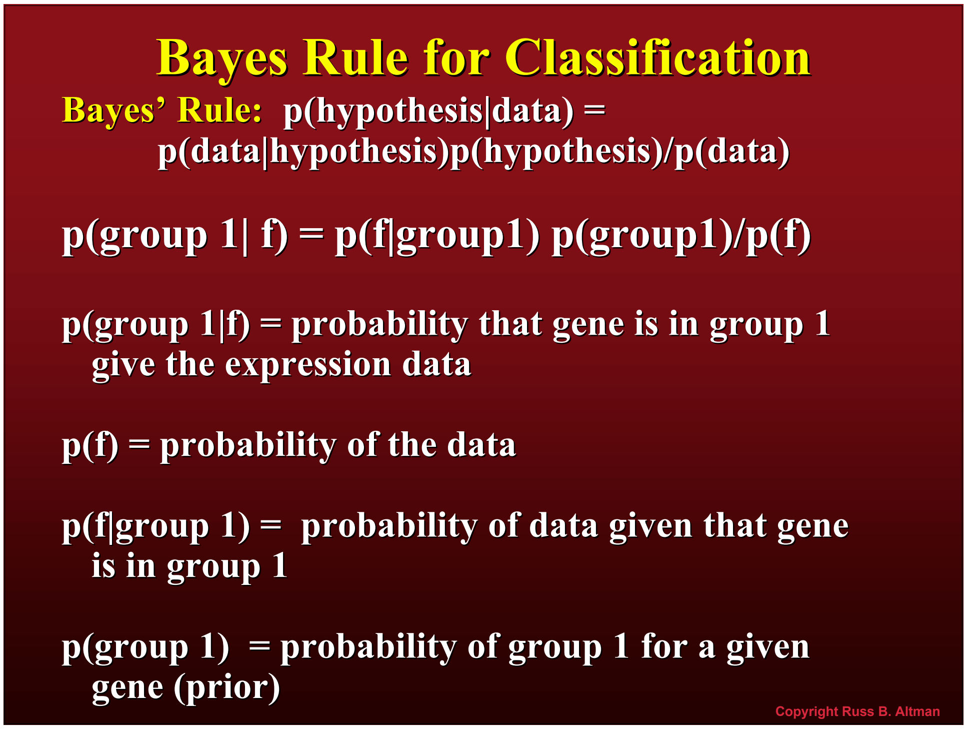
$p = e^{\beta f+a}/(1+e^{\beta f+a})$

Use optimization methods to find β (vector) that maximizes the difference between two groups. Then, can use equation to estimate membership of a gene in a group.



Logistic Model

PREDICT RED if high value for A and low value for B, (high weight on x coordinate, negative weight on y), but with Sigmoid transition from low prob to high prob.



Bayes Rule for Classification

Bayes' Rule: p(hypothesis|data) =
p(data|hypothesis)p(hypothesis)/p(data)

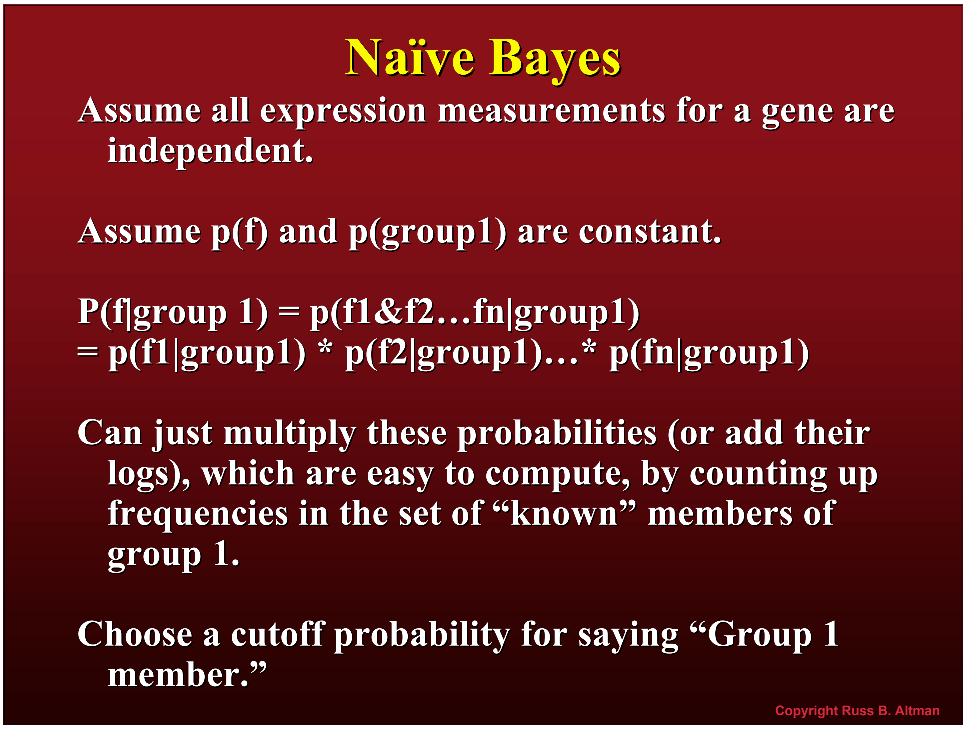
p(group 1| f) = p(f|group1) p(group1)/p(f)

p(group 1|f) = probability that gene is in group 1 give the expression data

p(f) = probability of the data

p(f|group 1) = probability of data given that gene is in group 1

p(group 1) = probability of group 1 for a given gene (prior)



Naïve Bayes

Assume all expression measurements for a gene are independent.

Assume p(f) and p(group1) are constant.

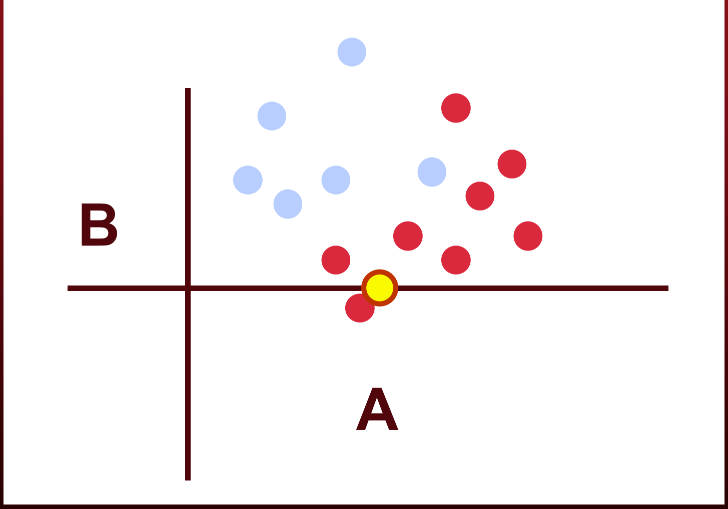
P(f|group 1) = p(f1&f2…fn|group1)
= p(f1|group1) * p(f2|group1)…* p(fn|group1)

Can just multiply these probabilities (or add their logs), which are easy to compute, by counting up frequencies in the set of "known" members of group 1.

Choose a cutoff probability for saying "Group 1 member."

Naïve Bayes

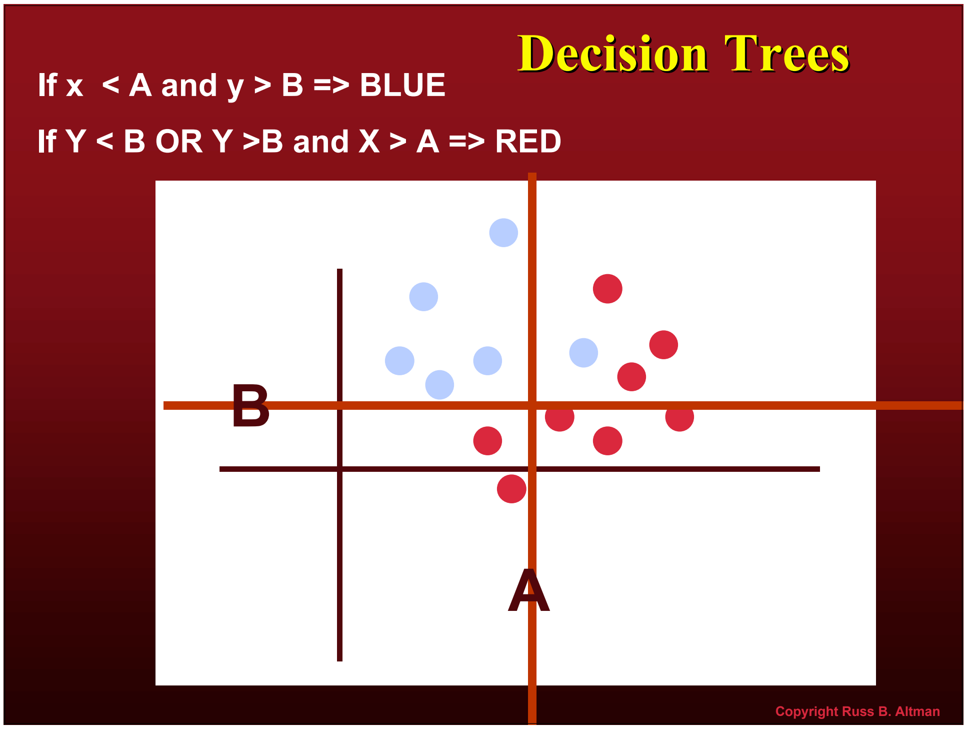
If P(Red|x=A) * P(Red| y = 0) = HIGH, so assign to RED



Decision Trees

Consider an n-dimensional graph of all data points (f, gene expression vectors).

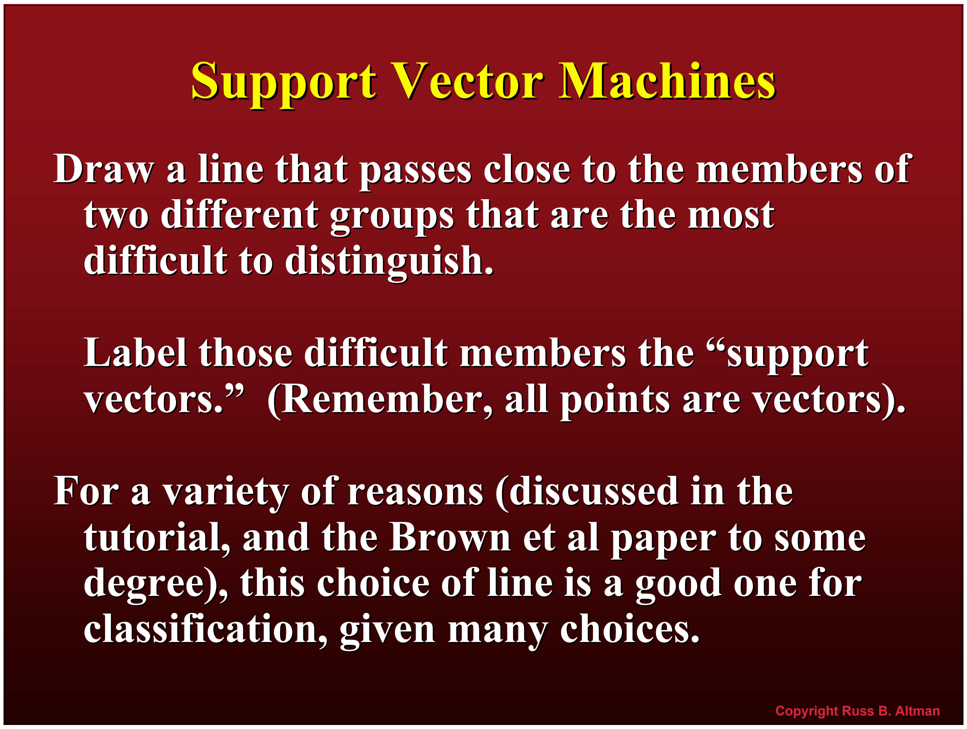
Try to learn cutoff values for each fi that separate different groups.



Decision Trees

If x < A and y > B => BLUE

If Y < B OR Y >B and X > A => RED

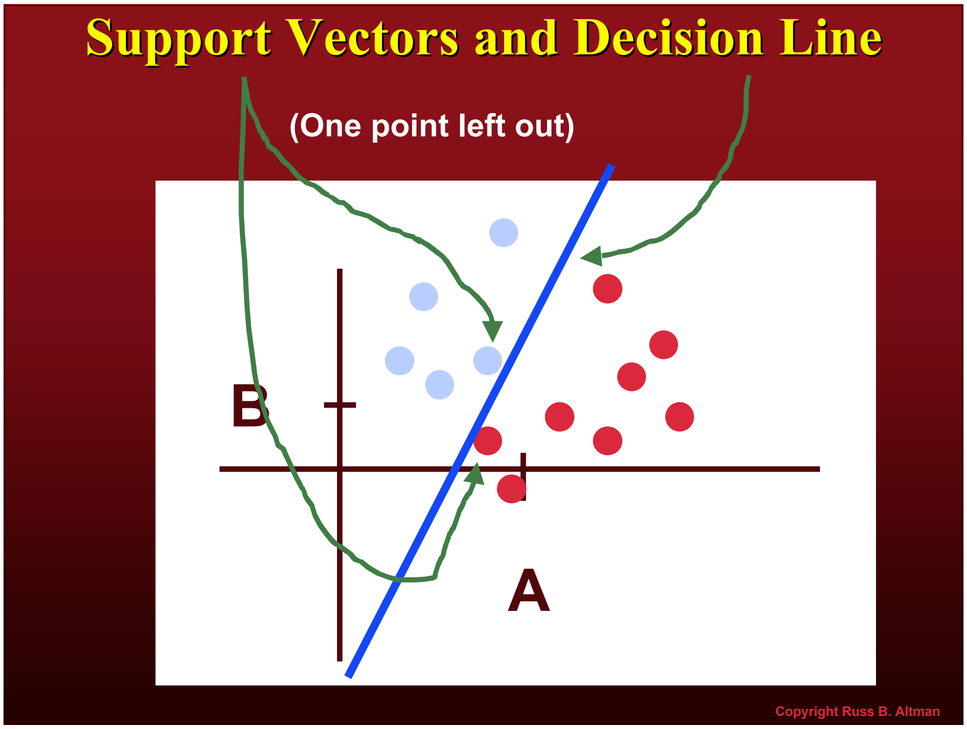


Support Vector Machines

Draw a line that passes close to the members of two different groups that are the most difficult to distinguish.

Label those difficult members the “support vectors.” (Remember, all points are vectors).

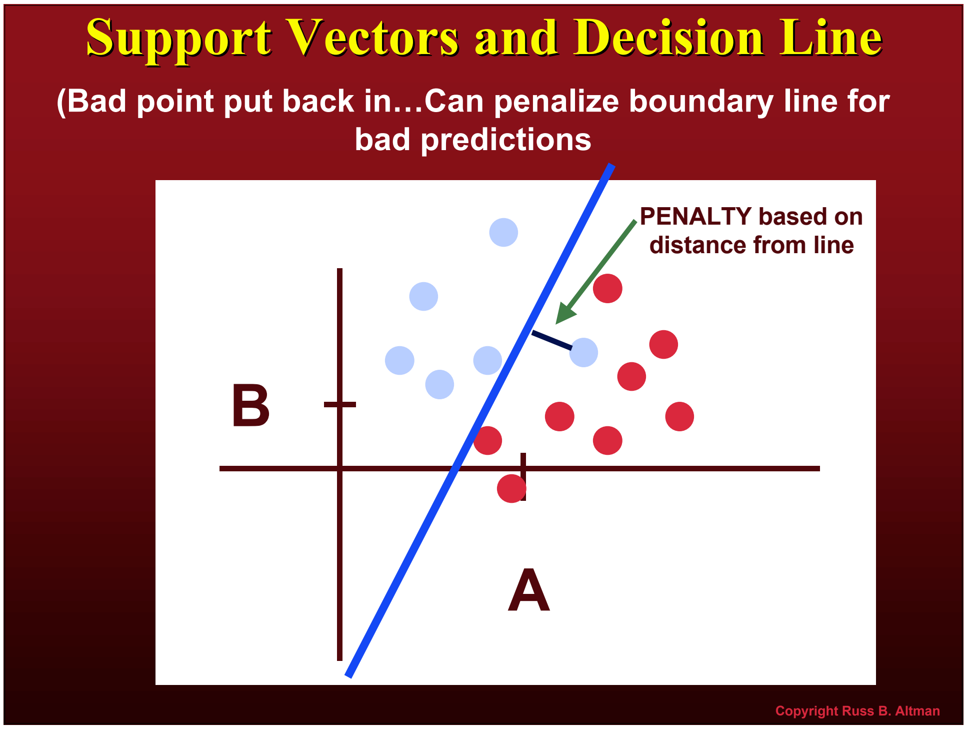
For a variety of reasons (discussed in the tutorial, and the Brown et al paper to some degree), this choice of line is a good one for classification, given many choices.



Support Vectors and Decision Line

(One point left out)

Copyright Russ B. Altman

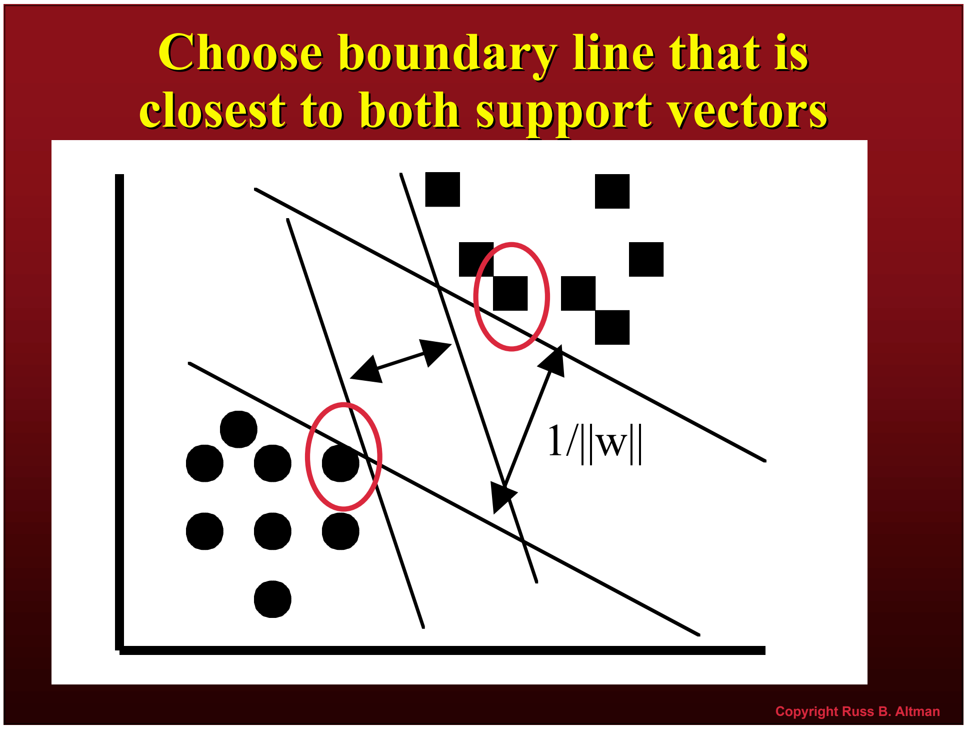


Support Vectors and Decision Line

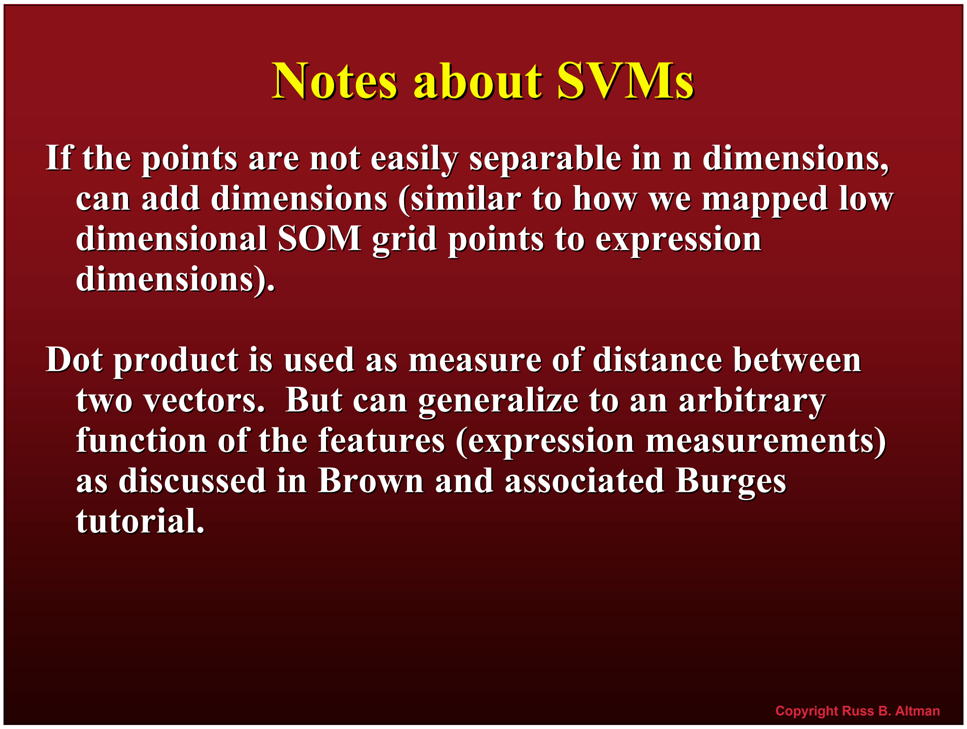
(Bad point put back in…Can penalize boundary line for bad predictions

PENALTY based on distance from line

Copyright Russ B. Altman



Choose boundary line that is closest to both support vectors



Notes about SVMs

If the points are not easily separable in n dimensions, can add dimensions (similar to how we mapped low dimensional SOM grid points to expression dimensions).

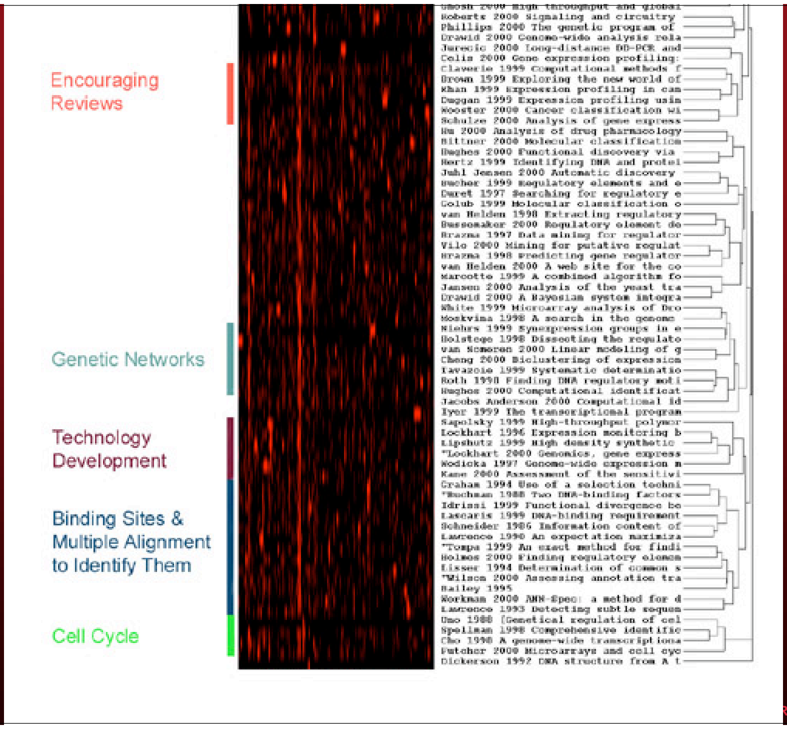
Dot product is used as measure of distance between two vectors. But can generalize to an arbitrary function of the features (expression measurements) as discussed in Brown and associated Burges tutorial.

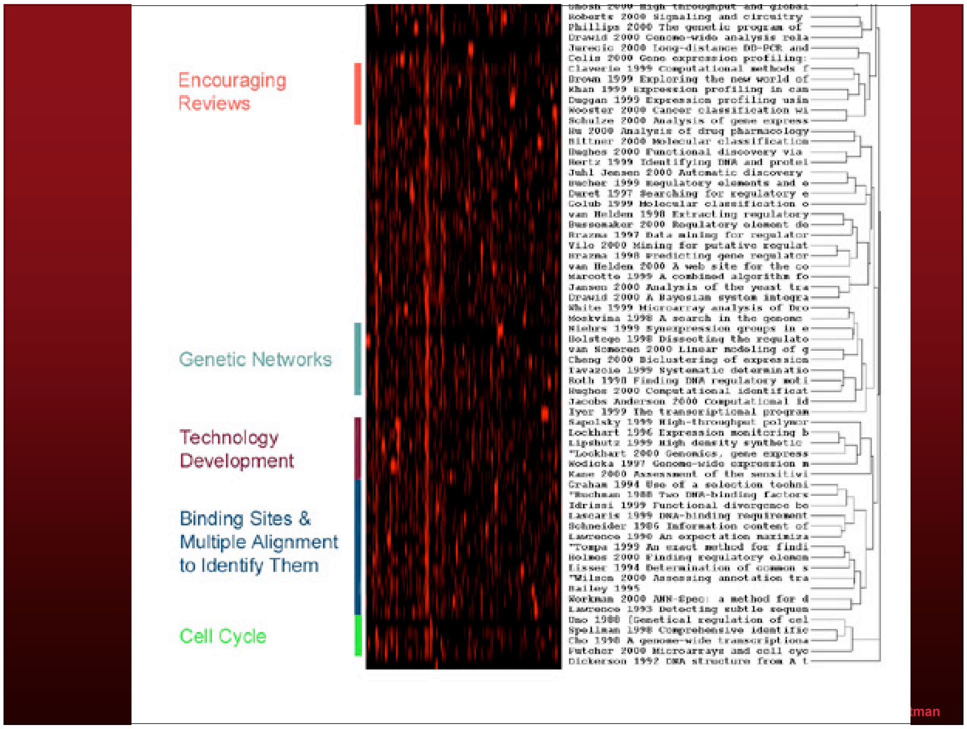
Evaluating Yes/No Classifiers

True Positives
False Positives
True Negatives
False Negatives

Sensitivity = TP/(TP + FN)
Specificity = TN/(TN + FP)
Positive Predictive Value = TP/(TP + FP)

ROC Curve = Plot Sensitivity vs. Specificity
(or Sensitivity vs. 1-Specificity)





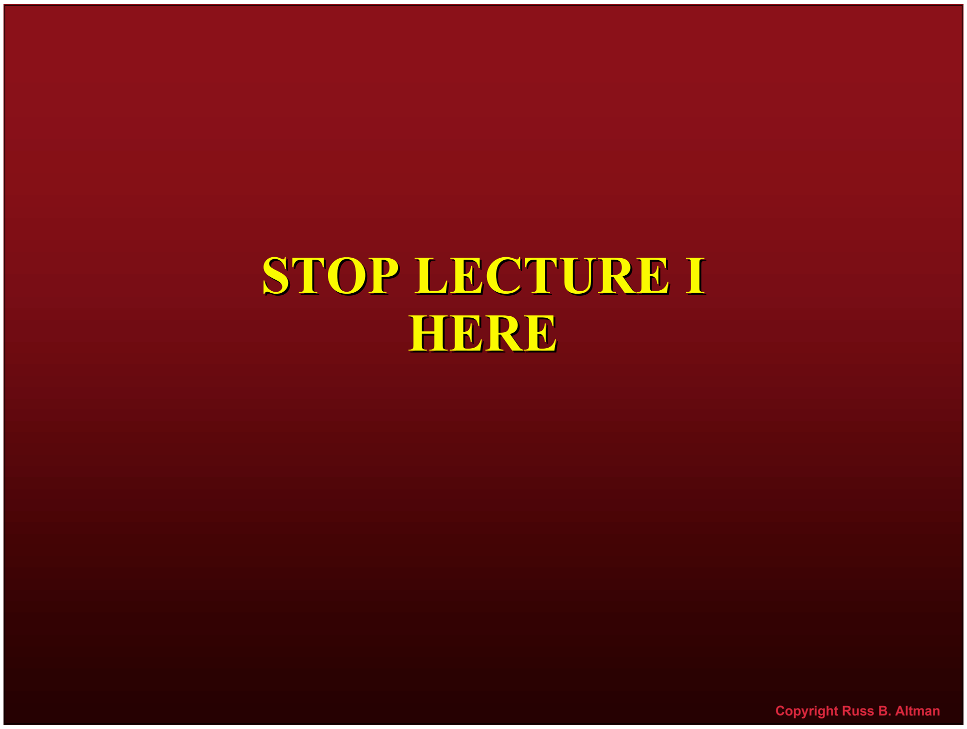
Encouraging Reviews

Genetic Networks

Technology Development

Binding Sites & Multiple Alignment to Identify Them

Cell Cycle



STOP LECTURE I HERE