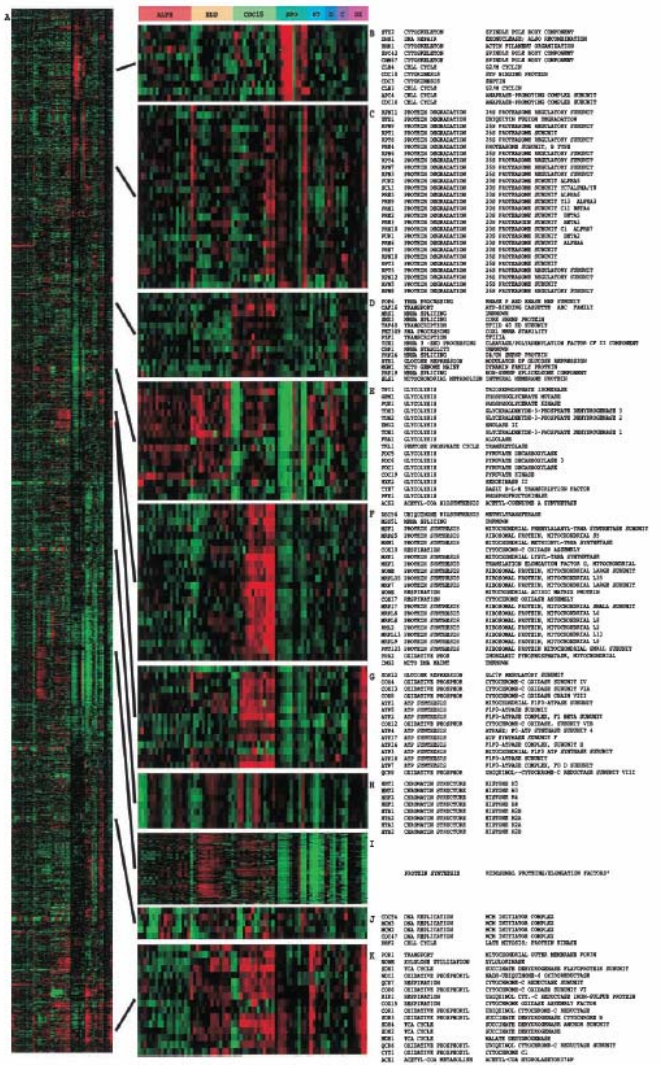
10-810: Advanced Algorithms and Models for Computational Biology

Optimal leaf ordering and classification

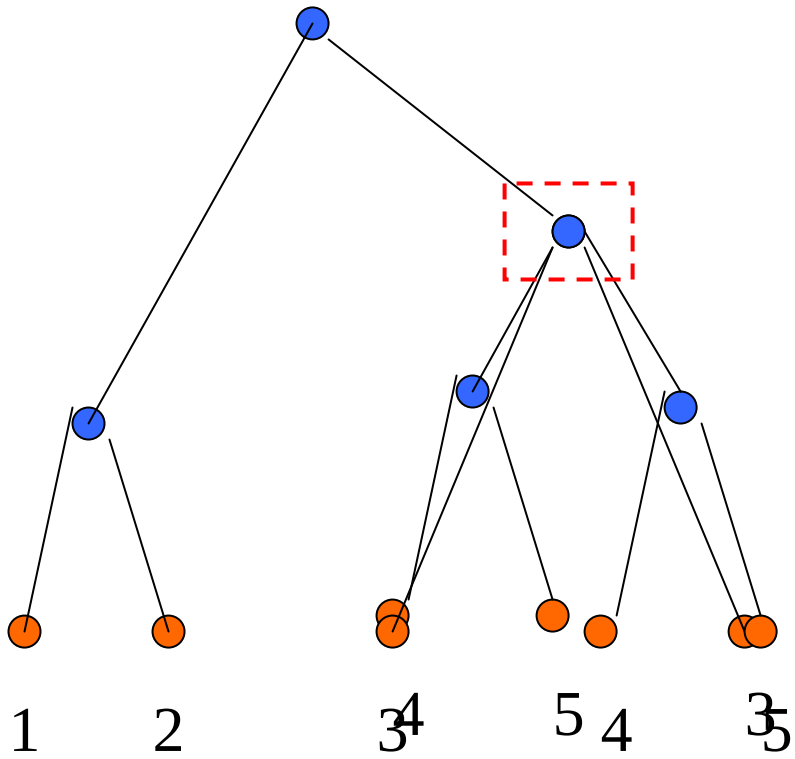
Hierarchical clustering

- As we mentioned, its one of the most popular methods for clustering gene expression data
- One of its main advantages is the global overview of the entire experiment in one figure.
- Biologists often omit the tree and use the figure to determine functional assignments



Clustering tree

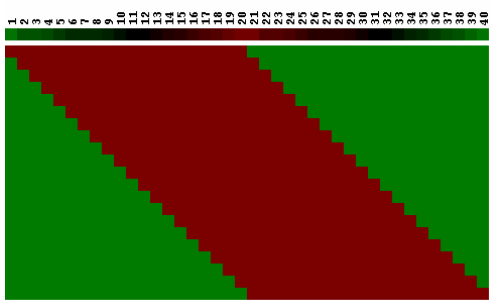
- For *n* leaves there are *n-1* internal nodes
- Each flip in an internal node creates a new linear ordering
- There are 2^{n-1} possible linear ordering of the leafs of the tree



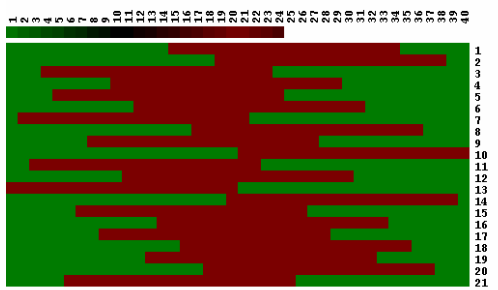
Importance of the Ordering

• Genes that are adjacent in the linear ordering are often hypothesized to share a common function.

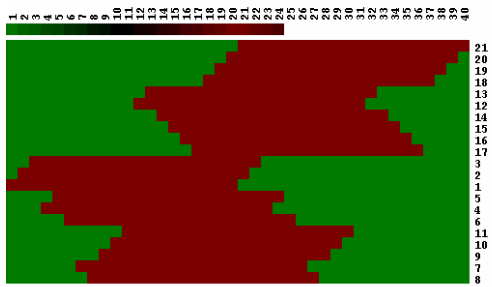
• Ordering can help determine relationships between genes and clusters in time series data analysis.



Initial structure



Permuted



Hierarchical clustering

Some heuristics

• Due to the large number of possible orderings (2^{n-1}), finding the optimal ordering was considered impractical by Eisen [Eisen98]

• Thus, some heuristics have been suggested for this problem:

- Order genes based on their expression levels [Eisen98]

- Order clusters using results of one dimensional som (Cluster)

- Order leaves and internal nodes based on similarity to parents siblings [Alon99]

Problem Definition

Denote by Φ the space of the possible linear orderings consistent with the tree.

Denote by $v_1 \ldots v_n$ the tree leaves.

Our goal is to find an ordering that maximizes the similarity of adjacent elements:

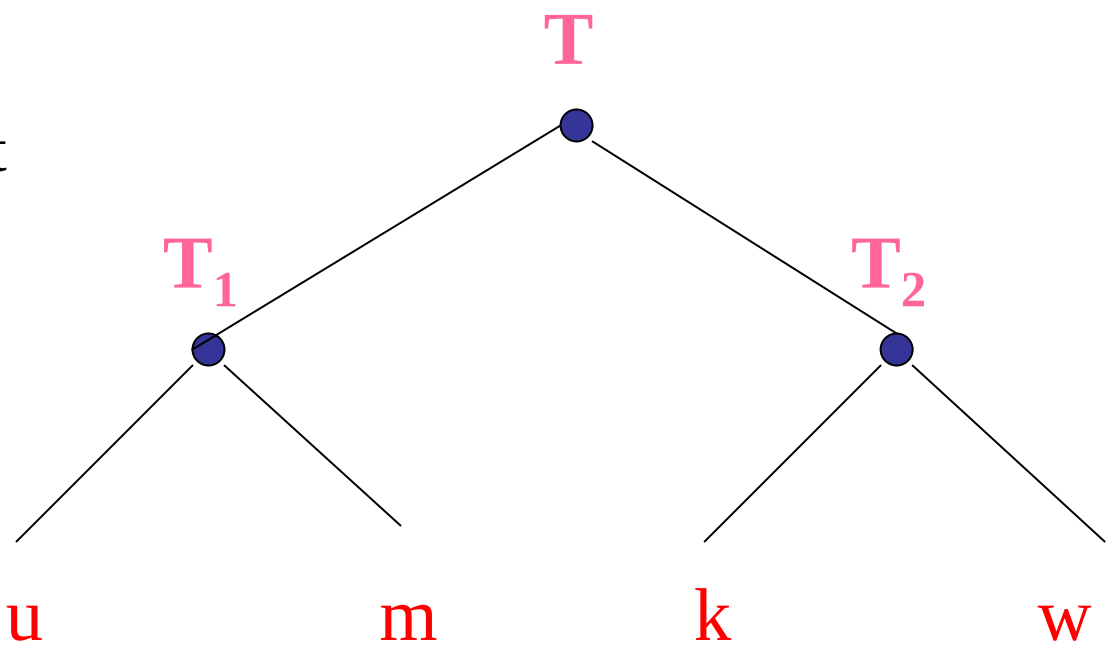
$$\max_{\phi \in \Phi} \sum_{i=1}^{n-1} S(v_i^{\phi}, v_{i+1}^{\phi})$$

where S is the similarity matrix

Computing the Optimal Similarity

Recursively compute the optimal similarity $L_T(u,w)$ for any pair of leaves (u,w) which could be on different **corners** (leftmost and rightmost) of T.

For a leaf $u \in T$, $C_T(u)$ is the set of all possible corner leaves of T when u is on one corner of T.

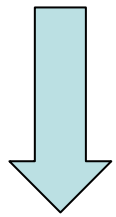


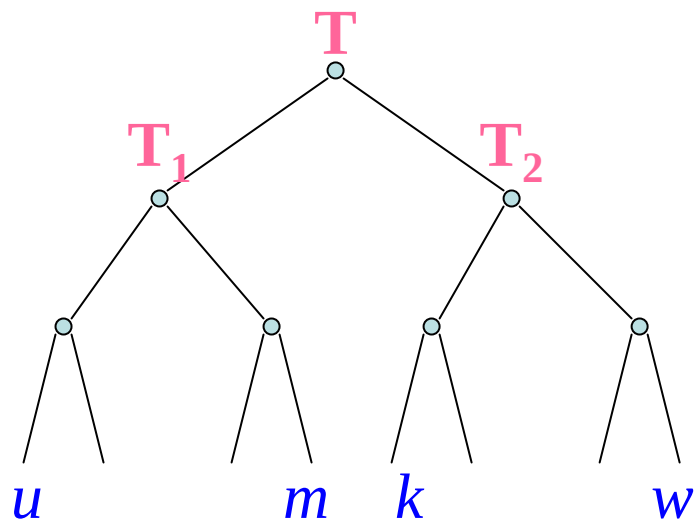
$$L_T(u,w) = \max_{m \in C_{T_1}(u), k \in C_{T_2}(w)} L_{T_1}(u,m) + L_{T_2}(k,w) + S(m,k)$$

For all $u \in T_1$

For all $w \in T_2$

$$L_T(u,w) = max_{m \in C_{T_1}(u), k \in C_{T_2}(w)} L_{T_1}(u,m) + L_{T_2}(k,w) + S(m,k)$$





For all $u \in T_1$

For all $k \in T_2$

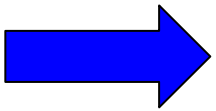
$$LL(u,k) = max_{m \in C_{T_1}(u)} L_{T_1}(u,m) + S(m,k)$$

For all $w \in T_2$

$$L_T(u,w) = max_{k \in C_{T_2}(w)} LL(u,k) + L_{T_2}(w,k)$$

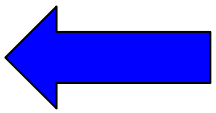
Algorithm Complexity

Time complexity : $F(n) = \theta(n^3)$



By induction. If $T = T_1, T_2$ and $|T| = n$, $|T_1| = s$ and $|T_2| = r$ we have:

$$F(n) \leq sr^2 + s^2r + F(s) + F(r) \leq (s+r)^3 \leq n^3$$



For the complete balanced binary tree with n leaves we have:

?

Space complexity:

We store one value for each pair of leaves. We use pointers to reconstruct the path we took. Thus, space complexity is $O(n^2)$.

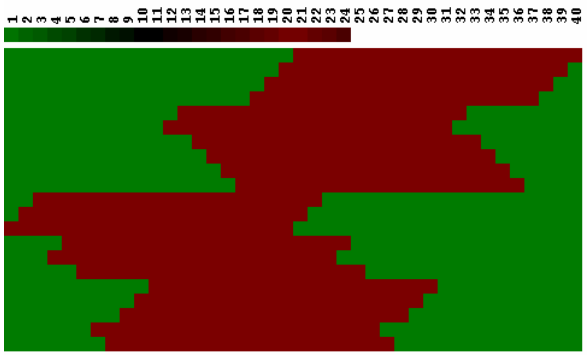
Random Inputs

Num of leaves	Num of values	Computing S	Improved $O(n^4)$	$O(n^3)$	Improved $O(n^3)$
400	20	0	2	2	1
900	30	2	20	25	3
1600	40	10	190	140	19
2500	50	29	900 (15 min)	580 (10 min)	59
3600	60	72	5700 (95 min)	1850 (30 min)	186

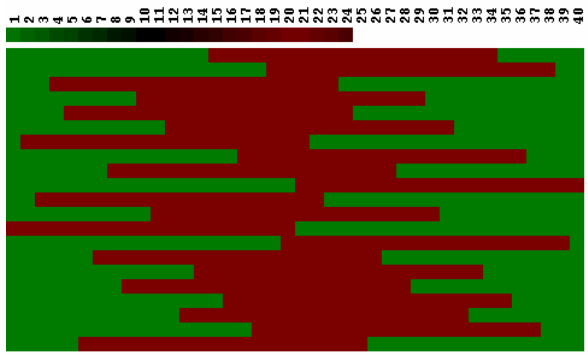
Running Time – Biological Datasets

type of dataset	num of genes	num of experiments	Computing *S*	Improved $O(n^4)$	$O(n^3)$	Improved $O(n^3)$
Cell cycle – cdc15	800	24	1	16	12	2
Cell cycle (Spellman)	800	59	3	14	12	1
Different sources (Eisen)	979	79	7	26	20	2
Environment response (Young)	3684	45	55	259	437	209

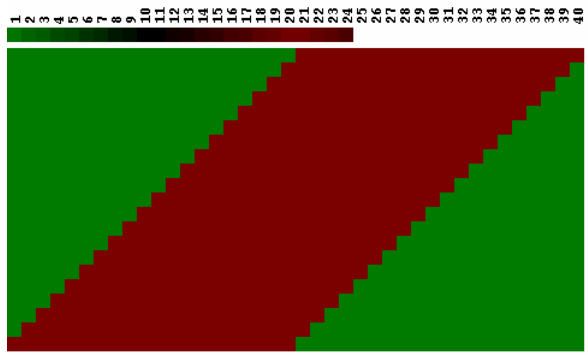
Results – Synthetic Data



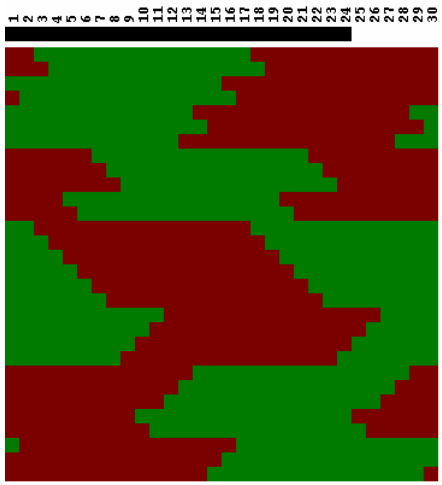
Hierarchical clustering



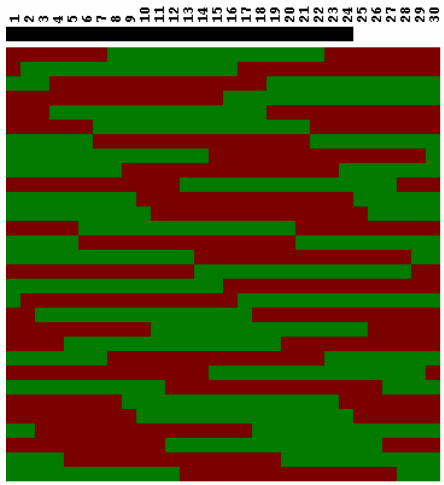
Input



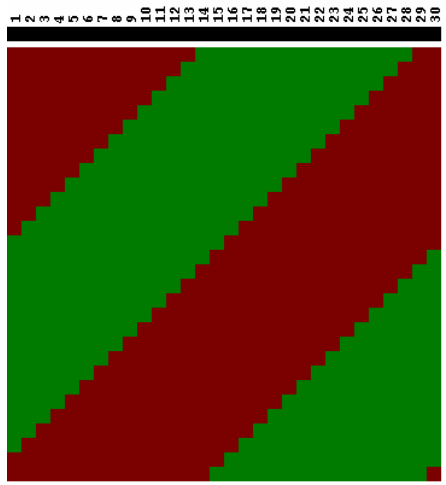
Optimal ordering



Hierarchical clustering



Input



Optimal ordering

Biological Results

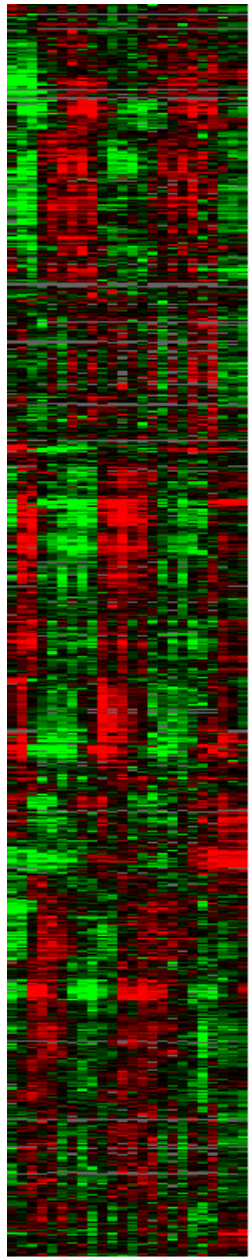
• Spellman identified 800 genes as cell cycle regulated in *Saccharomyces cerevisiae*.

• Genes were assigned to five groups termed *G1,S,S/G2,G2/M* and *M/G1* which approximate the commonly used cell cycle groups in the literature.

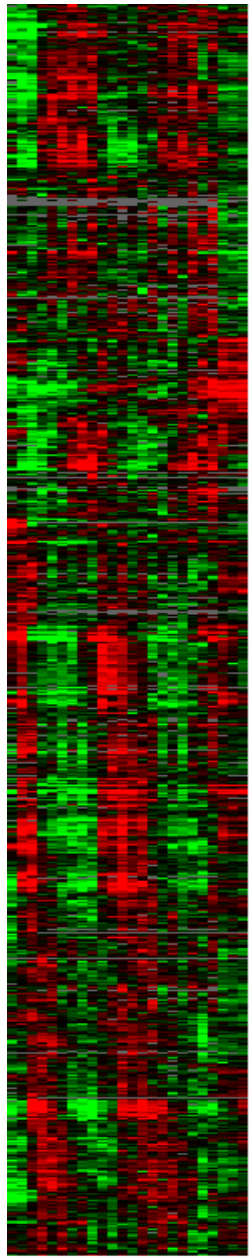
• This assignment was performed using a ‘phasing’ method which is a supervised classification algorithm.

• In addition to the phasing method, the authors clustered these genes using hierarchical clustering

Cell Cycle – 24 experiments of cdc15 temperature sensitive mutant

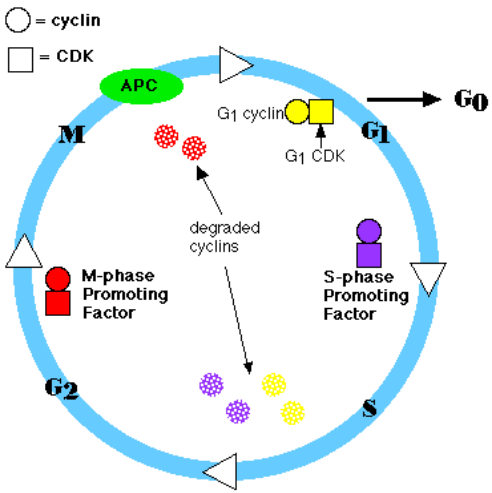


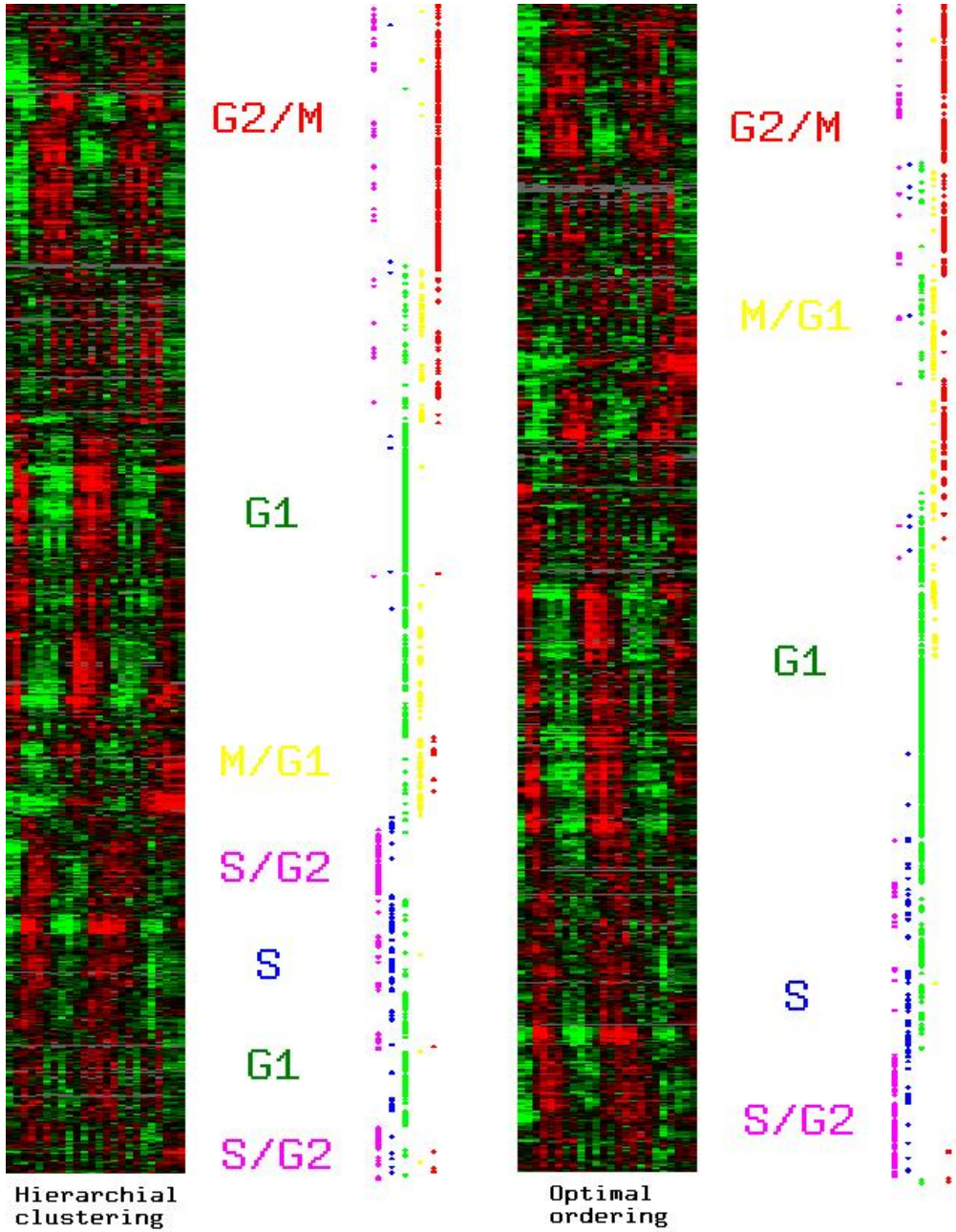
Hierarchical clustering



Optimal ordering

24 experiments of cdc15 temperature sensitive mutant





Classification

Types of classifiers

- We can divide the large variety of classification approaches into roughly two main types

1. Generative:
   - build a generative statistical model
   - e.g., mixture model

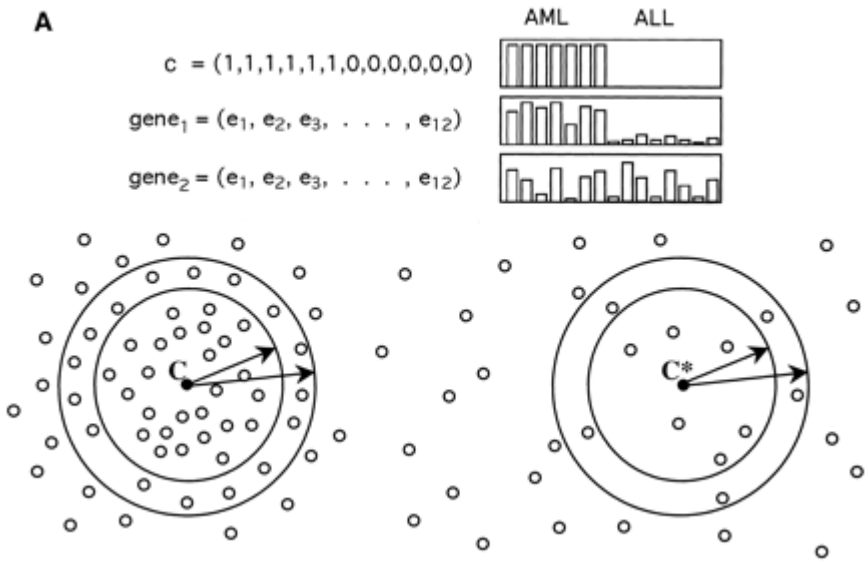
2. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., logistic regression

Golub et al

- 38 test samples (27 ALL 11 AML)
- Each gene was initially compared to an idealized expression pattern: 11111111111111000000000000000000 for class 1 and similarly 00000000000000000000011111111111111 for the second class.
- The actual selection was done by setting:

$$p(g,c) = \frac{\mu_1(g) - \mu_2(g)}{\sigma_1(g) + \sigma_2(g)}$$

- Large values of |p(g,c)| indicate strong correlation between the gene and the classes, and the sign of p(g,c) depends on the class in which this gene is expressed.


A
AML
ALL
c = (1,1,1,1,1,1,1,0,0,0,0,0,0)
gene1 = (e1, e2, e3, . . . , e12)
gene2 = (e1, e2, e3, . . . , e12)
C
C*

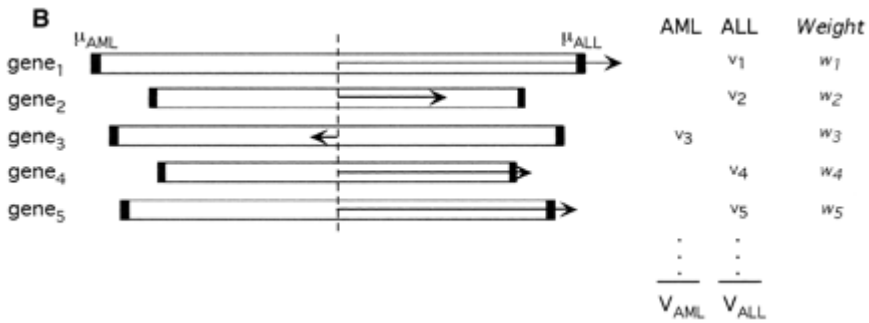
Weighted voting

- Use a subset of the selected genes (50).
- Set $a_g = p(g,c)$ and $b_g = (\mu_1(g)+\mu_2(g))/2$
- Given a new sample X, we set the vote of gene g to:

$$v_g = a_g(x_g - b_g)$$

- A positive value is a vote for class 1 and a negative for the second class

Weighted voting



Voting strength

- The votes are summed for each of the two classes.
- The decision is made by using:

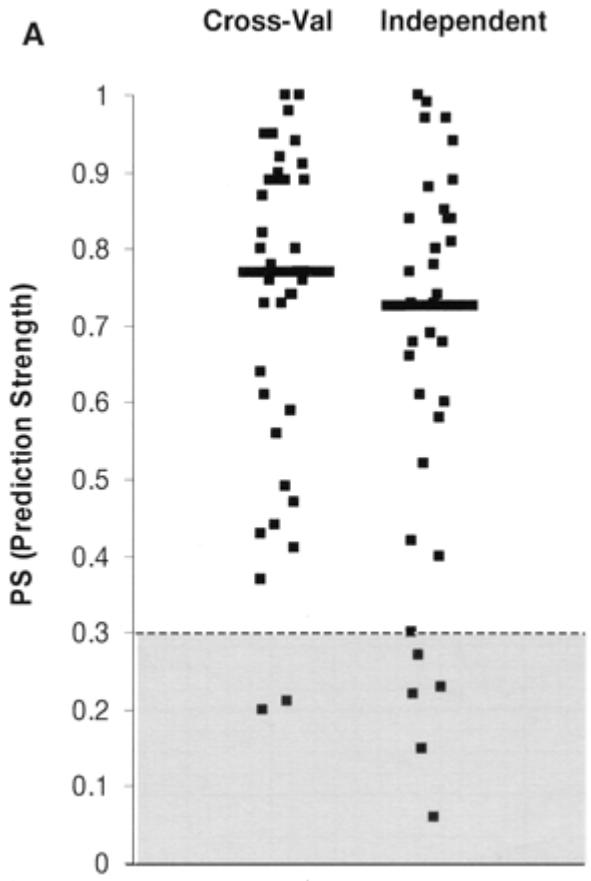
$$PS = \frac{v_{win} - v_{lose}}{v_{win} + v_{lose}}$$

- PS determines our confidence in the classification result.
- How do we chose PS ?

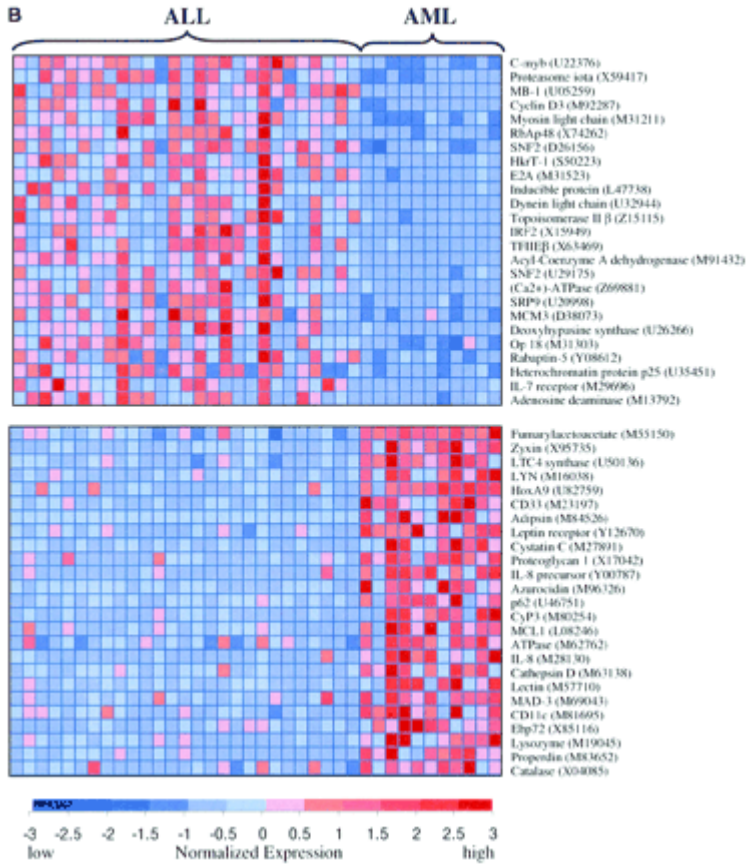
Testing the classifier

- Cross validation.
- Test set: 38 samples:
  - 20 ALL
  - 14 AML
- 29 of 34 had a classification value higher than the threshold and all were predicted correctly.

Classification results



Selected genes



Can we do better?

Generative classifiers

- A mixture of two Gaussians, one Gaussian per class choice of class:

$$X \in class \quad 1 \Rightarrow X \sim (\mu_1, \sigma_1)$$
$$X \in class \quad 0 \Rightarrow X \sim (\mu_0, \sigma_0)$$

- where X corresponds to, e.g., a tissue sample (expression levels across the genes).

- Three basic problems we need to address:
  - decisions
  - estimation
  - variable (feature) selection

Decision: Bayesian classifiers

- Given a probabilistic model and an unlabeled data vector **X**, we can use Bayes rule to determine the class:

$$p(class = 1 \mid X) = \frac{P(X \mid class = 1)P(class = 1)}{P(X \mid class = 1)P(class = 1) + P(X \mid class = 0)P(class = 0)}$$

- We compute *p(class=1|X)* and *p(class =0|X)* and chose the class with the highest probability
- This method can be easily extended to multiple classes

Decision boundary

- Given a probabilistic model and an unlabeled data vector **X**, we can use Bayes rule to determine the class:

$$p(class = 1 | X) = \frac{P(X | class = 1) P(class = 1)}{P(X | class = 1) + P(X | class = 0)}$$

- Using Bayes classifiers, the decision comes down to the following (log) likelihood ratio:

$$\log \frac{p(X | \mu_1, \sigma_1) p(class = 1)}{p(X | \mu_0, \sigma_0) p(class = 0)} > 0 \Rightarrow class = 1$$

Decision boundary

- Using Bayes classifiers, the decision comes down to the following (log) likelihood ratio:

$$\log \frac{p(X \mid \mu_1, \sigma_1)\, p(class = 1)}{p(X \mid \mu_0, \sigma_0)\, p(class = 0)} > 0 \Rightarrow class = 1$$

Why?

The prior class probabilities P(class) bias our decisions towards one class or the other.

- Decision boundary:

$$\log \frac{p(X \mid \mu_1, \sigma_1)\, p(class = 1)}{p(X \mid \mu_0, \sigma_0)\, p(class = 0)} = 0$$

Decision boundaries

- Equal covariances

$$X \sim (\mu_1, \Sigma\ ); class = 1$$

$$X \sim (\mu_0, \Sigma\ ); class = 0$$

The decision rule is linear

Decision boundaries

- Unequal covariances

$$X \sim (\mu_1, \sigma_1); class = 1$$
$$X \sim (\mu_0, \sigma_0); class = 0$$

- The decision rule is quadratic

Estimation

- Suppose we are given a set of labeled tissue samples

  $X^1 \ldots X^k$ – class = 1

  $X^{k+1} \ldots X_n$ – class =0

- We can estimate the two Gaussians separately.
- For example, maximum likelihood estimation gives

  $P(class=1) = k/n$

  μ_1 = sample mean of $X^1 \ldots X^k$

  Σ_1 = sample covariance of $X^1 \ldots X^k$
- and similarly for the other class(es)
- We already mentioned that this is the MLE estimator

Golub et al

- Leukemia classification problem

- 7130 ORFs (expression levels)
- 38 labeled training examples,
- 34 test examples

Our mixture model (assume equal class priors)

$$X \sim (\mu_1, \Sigma\ ); class = 1$$

$$X \sim (\mu_0, \Sigma\ ); class = 0$$

Problems?

Golub et al

•Leukemia classification problem

•7130 ORFs (expression levels)

•38 labeled training examples,

•34 test examples

Our mixture model (assume equal class priors)

$$X \sim (\mu_1, \Sigma\ ); class = 1$$

$$X \sim (\mu_0, \Sigma\ ); class = 0$$

Problems?

For 7000+ genes we would need to set roughly 18,000,000 parameters in each covariance matrix! (with 38 examples)

Naïve Bayes classifiers

- This full covariance model is too complex, we need to constrain the covariance matrices
- The simplest constraint we can use is a diagonal covariance matrix instead of a full covariance
- When using such a matrix we make the (implicit) assumption that the genes are *independent* given the class labels
- In other words, we assume that:

$$p(X \mid class = 1) = \prod_i p(X_i \mid class = 1)$$

$$X_i \sim N(\mu_i^1, \sigma_i^2)$$

where X_i is the value for gene i

Naïve Bayes classifiers

- Lets further assume equal variance for a specific gene across the two sets of samples (that is, noise is independent of the sample condition)
- As a result, we need to only estimate class-conditional means and a common variance for each gene
- How well might we do in the Golub et al. task?

3 test errors (out of 34)

Feature selection

- Test which genes are predictive of the class distinction
- Why is this important? Is more information always better?
- We can test the predictive power of genes by testing if the mean expression level is different in the two class populations
- We assume the two classes (0 and 1) have the same covariance matrix

Feature selection

- H_0 is that a gene is not predictive of the class label
- H_1 is that a gene can predict the class label

$$H_0 = X_1 \sim N(\mu, \sigma^2), X_2 \sim N(\mu, \sigma^2)$$
$$H_1 = X_1 \sim N(\mu_1, \sigma^2), X_2 \sim N(\mu_2, \sigma^2)$$

- We can use a likelihood ratio test for this purpose Let x_i^t denote the observed expression levels for gene i
- The parameter estimates are computed from the available populations in accordance with the hypothesis.

Gene selection (cont.)

- We rank the genes in the descending order of the test statistics $T(x_i)$.
- How many genes should we include?
- We include all the genes for which the associated p-value of the test statistic is less than 1/m, where m is the number of genes

- This ensures that we get on average only 1 erroneous predictor (gene) after applying the test for all the genes

Golub example

- In the Golub et al. problem, we get 187 genes, and only 1 test error (out of 34)
- How many genes do we really need?
- Only a few genes are necessary for making accurate class distinctions

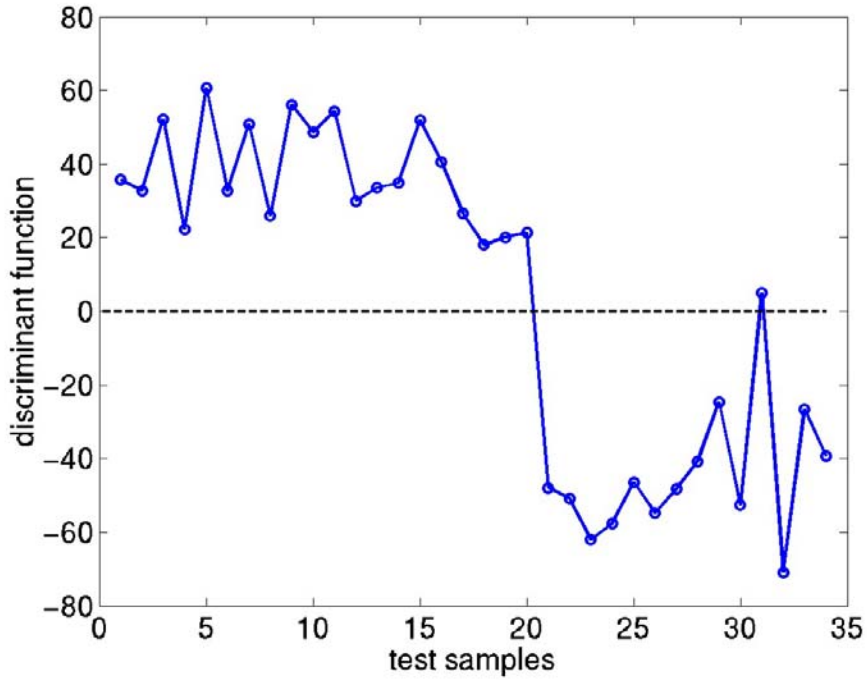
Golub cont.

- The figure shows the value of the discriminant function

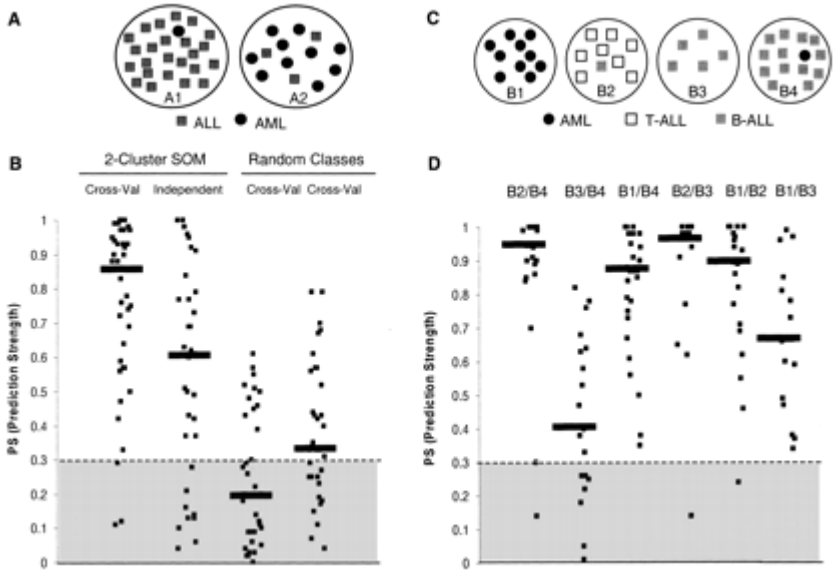
$$f(x) = \log \frac{p(X \mid \mu_1, \sigma_1)}{p(X \mid \mu_0, \sigma_0)}$$

across the test examples

- The only test error is also the decision with the lowest confidence



Unsupervised



- Build a class predictor using the clustering algorithm
- Use cross validation to determine class membership
- Problems ?

What you should know

- Optimal ordering can help interpreting expression results
- Different classifier types
- Cross validation, feature selection