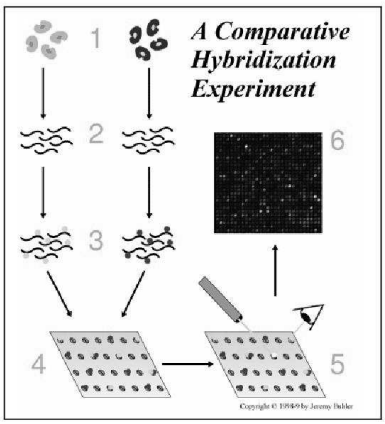
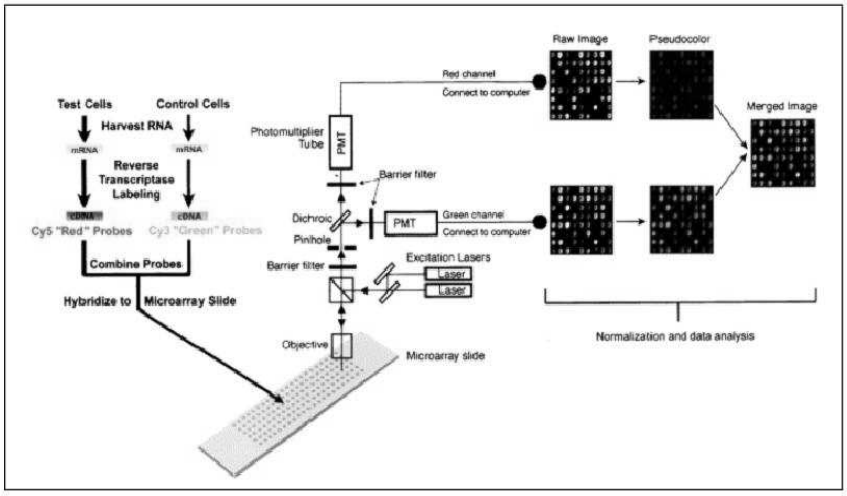
CISC 889 Bioinformatics (Spring 2004)

DNA Microarray and Gene Expression

- Gene expression
  - How many copies of a gene (its product) is present in the cell?
  - For experimental reasons, gene expressions are measured by numbers of mRNAs, not directly by proteins. (See Proteomics)
  - Various cell types are due to different genes expressed.
  - The difference between diseased (e.g., cancerous) and non-diseased
  - Diseased cells are often resulted from the abnormal levels of expression of key genes.



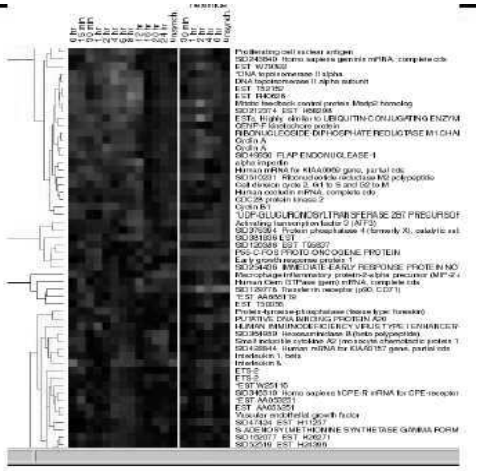
- Microarray
  - Oligonucleotide (Affymetrix) array
    - Oligo (~ 25 bases long)
    - High density (1cm^2 contain 100k oligos)
  - cDNA array
    - cDNA (RT-PCR), much longer (> 1000 bases)
    - Varied density of cDNA on each spot, hybridization depends on length
    - Less possibility for false positives
  - Image processing
  - Background subtraction
  - Normalization


Test Cells
Control Cells
Harvest RNA
mRNA
mRNA
Reverse Transcriptase Labeling
cDNA
cDNA
Cy5 "Red" Probes
Cy3 "Green" Probes
Combine Probes
Hybridize to Microarray Slide
Photomultiplier Tube
PMT
Barrier filter
Dichroic
PMT
Pinhole
Barrier filter
Excitation Lasers
Laser
Laser
Objective
Microarray slide
Red channel
Connect to computer
Green channel
Connect to computer
Raw Image
Pseudocolor
Merged Image
Normalization and data analysis

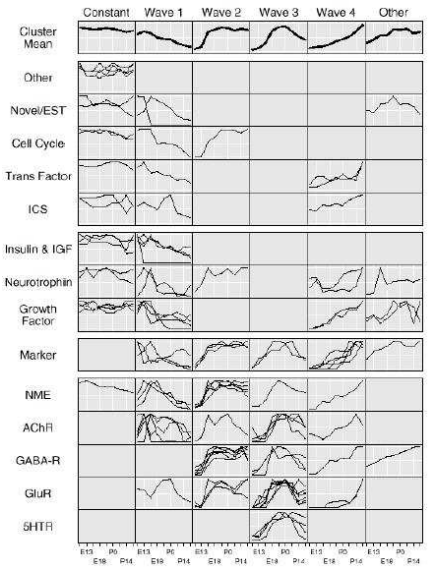

JOURNAL OF BACTERIOLOGY
AUGUST 2000, VOLUME 182, NUMBER 16, PAGES 4375–4676
Journal of Bacteriology
AMERICAN SOCIETY FOR MICROBIOLOGY
AUGUST 2000, VOLUME 182, NUMBER 16
Published Twice Monthly

Gene Expression Data Analysis

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

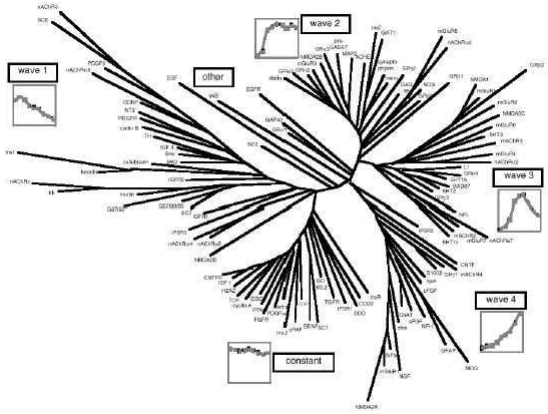


Russ Altman



Russ Altman

Gene Expression Data Analysis



Russ Altman

Applications

- Understanding correlation b/w genotype and phenotype
- predicting genotype <=> phenotype
- Phenotypes:
  - drug/therapy response
  - drug-drug interactions for expression
  - drug mechanism
  - interacting pathways of metabolism

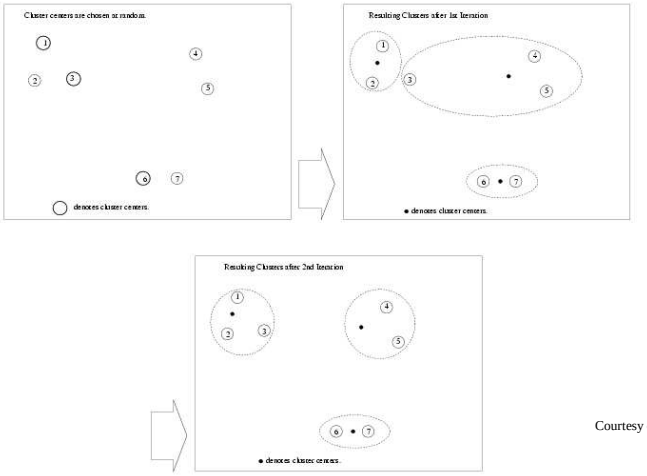
Iterative Distance-based Clustering (K-means)

Basic idea: Given a predetermined constant k (the number of clusters), iteratively recompute centers (means) of k clusters starting from randomly chosen k instances as centers.

1. K instances are chosen at random as cluster centers.
2. Instances are assigned to their closest cluster center, generating k cluster.
3. **while** (there is change in cluster centers)
4. Compute the centroid (mean) of all instances in each cluster.
5. Instances are assigned to their closest cluster center, generating k cluster.
6. **end**

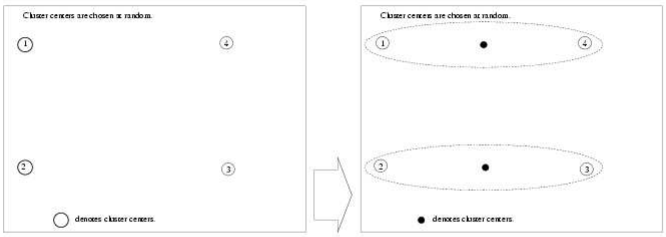
Courtesy of Sun Kim

A Correct Clustering Example



Courtesy of Sun Kim

An Incorrect Clustering Example



The initial choice of cluster centers, node 1 and node2, leads to an incorrect clustering. Obviously. a different choice of cluster centers, node 1 and node 3, result in a correct clustering.

Courtesy of Sun Kim

Discussion

1. The iterative procedure for k-means may end up with a local minimum, depending on the initial choice for cluster centers.
2. A simple heuristic is to run the k-mean clustering several times with different starting points.
3. How do we know the number of clusters in advance?
   Many different k can be tried.
4. K-mean clustering, as most clustering techniques, assumes that instances can be placed in Euclidian space.
5. Speeding up the K-mean algorithm is important.
   See the paper in SIGKDD Exploration (July 2000) by Farnstorm, Lewis, and Elkan.
   `http://www-cse.ucsd.edu/~elkan`

Courtesy of Sun Kim

CLICK (by Ron Shamir)

CLICK (CLuster Identification via Connectivity Kernels) is a newer algorithm for clustering [20]. The input for CLICK is the gene expression matrix. Each row of this matrix is an "expression fingerprint" for a single gene. The columns are specific conditions under which gene expression is measured (e.g. different points in time). A more formal definition is as follows:

Let $N = \{e_1, \ldots, e_n\}$ be a set of elements. Let M be an input real-valued matrix of order $n \times p$, where M_{ij} is the j-th attribute of e_i. The i-th row-vector in M is the fingerprint of e_j. For a set of elements $K \subseteq N$, we define the *fingerprint* of K to be the mean vector of the fingerprints of the members of K. One seeks to partition N into clusters (subsets). In such a partition, elements in the same cluster are called *mates*.

The CLICK algorithm attempts to find a partition of N into clusters, so that two criteria are satisfied: *Homogeneity* - mates are highly similar to each other; and *separation* - non-mates have low similarity to each other.

CLICK (by Ron Shamir)

Probabilistic Assumptions

The CLICK algorithm makes the following assumptions:

1. Similarity values between mates are normally distributed with mean μ_T and variance σ_T^2.
2. Similarity values between non-mates are normally distributed with mean μ_F and variance σ_F^2.
3. $\mu_T > \mu_F$

These assumptions are justified both empirically and theoretically by the Central Limit Theorem.

CLICK (by Ron Shamir)

The Basic CLICK Algorithm

The CLICK algorithm represents the input data as a weighted *similarity graph* $G = (V, E)$. In this graph vertices correspond to elements and edge weights are derived from the similarity values. The weight w_{ij} of an edge (i, j) reflects the probability that i and j are mates, and is set to be

$$w_{ij} = \log \frac{p_{mates} f(S_{ij}|i,j \text{ are mates})}{(1 - p_{mates}) f(S_{ij}|i,j \text{ are non-mates})}$$

where $f(S_{ij}|i,j \text{ are mates}) = f(S_{ij}|\mu_T, \sigma_T)$ is the value of the probability density function for mates at S_{ij}:

$$f(S_{ij}|i,j \text{ are mates}) = \frac{1}{\sqrt{2\pi}\sigma_T} e^{-\frac{(S_{ij}-\mu_T)^2}{2\sigma_T^2}}$$

Similarly, $f(S_{ij}|i,j \text{ are non-mates})$ is the value of the probability density function for non-mates.

CLICK (by Ron Shamir)

The idea behind the algorithm the following: given a connected graph G, we would like to decide whether $V(G)$ is a subset of some true cluster, or $V(G)$ contains elements from at least two true clusters. In the first case we say that G is *pure*. In order to make this decision we test for each cut C in G the following two hypotheses:

- H_0^C: C contains only edges between non-mates.
- H_1^C: C contains only edges between mates.

G is declared a *kernel* if H_1 is more probable for all cuts.

CLICK (by Ron Shamir)

Lemma 11.6 *G is a kernel iff $MinWeightCut(G) > 0$.*

Proof Using Bayes Theorem, it can be shown that

$$W(C) = \log \frac{Pr(H_1^C|C)}{Pr(H_0^C|C)}$$

Obviously, $W(C) > 0$ iff $Pr(H_1^C|C) > Pr(H_0^C|C)$. If the minimum cut is positive, then obviously so are all the cuts. Conversely, if the minimum cut is non-positive, then for that cut $Pr(H_1^C|C) \leq Pr(H_0^C|C)$, therefore G is not a kernel. ■

CLICK (by Ron Shamir)

```
Basic-CLICK(G(V, E))
    if (V(G) = {v}) then
        move v to the singleton set R
    elseif (G is a kernel) then
        Output V(G)
    else
        (H, H̄, cut) ← MinWeightCut(G)
        Basic-CLICK(H)
        Basic-CLICK(H̄)
    end if
end
```

CLICK (by Ron Shamir)